Reconstruction of 3D Building Models from Aerial Images and Maps

Ildikó Süveg

NCG Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission

Delft, May 2003

Reconstruction of 3D Building Models from Aerial Images and Maps
Ildikó Süveg
Publications on Geodesy 53
ISBN 90 6132 280 4
ISSN 0165 1706

Published by: NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission, Delft, The Netherlands
Printed by: Optima Grafische Communicatie, Optima Graphic Communication, Rotterdam, The Netherlands
Cover: Examples of reconstructed models of complex buildings

NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission
P.O. Box 5030, 2600 GA Delft, The Netherlands
Tel.: +31 (0)15 278 28 19
Fax: +31 (0)15 278 17 75
E-mail: ncg@citg.tudelft.nl
Website: www.ncg.knaw.nl

The NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission is an institute of the Royal Netherlands Academy of Arts and Sciences (KNAW)

Abstract

Reconstruction of 3D Building Models from Aerial Images and Maps

The 3D reconstruction of buildings has numerous applications in areas that include urban planning, construction, environment, communication, transportation, energy and property management, tourism, and virtual tours of cities. In this thesis, the reconstruction of 3D building models from aerial images is addressed. The approach presented in this thesis integrates the aerial images analysis with information from a GIS database and domain knowledge.

The problem of automatic 3D building reconstruction has been a central research topic in computer vision and image understanding communities as well as in digital photogrammetry for many years. A variety of approaches has been suggested for the reconstruction of buildings from aerial images. Despite considerable research effort, there is no complete system that can reliably perform autonomous 3D building reconstruction in a wide variety of scene domains. This is particularly true in complex urban areas containing buildings with different shapes and roof types as well as in complicated underlying terrain. Of course, some progress has been made, but there is room for improvement. This improvement can be achieved by fusing multiple data sources and some a priori information.

In this project, large-scale 2D GIS databases were used as additional information source. Combination of image data and map data turned out to improve the reliability of the reconstruction. Generic knowledge about the shape of the buildings is also incorporated in the system. Since most buildings can be described as an aggregation of simple building types, the knowledge about the problem domain can be represented in a building library containing simple building models. Therefore, a building library was defined containing the most common building primitives, such as flat roof, and different types of gable roofs.

The building reconstruction process was formulated as a multi-level hypothesis generation and verification scheme and it was implemented as a search tree. A method that can localize the buildings in images using map information has been developed. Also, a method for generating building hypotheses corresponding to the primitives defined in the building library has been developed. This implies stereo matching of image features (corners, lines) which correspond to map primitives and fitting of the building hypotheses to images.

A further contribution is the definition of a metric for evaluating the generated building hypotheses in order to select the one which best describes the image. The metric is based on the formulation of the mutual information between the building model and the images. Methods for the estimation of the mutual information from training samples were analyzed. This metric has been rigorously derived from information theory and does not require a priori information about the surface properties of the object and is robust with respect to variations of illumination. Also, no assumption about the shape of the objects are made. As result the method is quite general and may be used in a wide variety of applications.

The produced approach is able to meet most of the requirements of an automatic 3D building reconstruction system. The developed system has been used in urban and suburban areas to reconstruct buildings and showed good results. Experiments were carried out on two data sets with different characteristics. The system was able to reconstruct more than 80% of the buildings and the accuracy of the reconstruction is good enough for mapping purposes.

Ildikó Süveg, 2003

Samenvatting

Reconstructie van 3D gebouw modellen op basis van luchtfoto's en topografischekaarten

3D gebouwmodellen kennen talrijke toepassingen, o.a. op het gebied van stedebouw, bouw, milieu, communicatie, transport, energie, bouwmanagement, toerisme, en virtuele rondleidingen door steden. Dit proefschrift behandelt de reconstructie van 3D gebouwmodellen op basis van luchtfoto's. De methode die in dit proefschrift wordt voorgelegd integreert de analyse van luchtfoto's, informatie uit een GIS database en generieke kennis over gebouwen.

Het komen tot automatische 3D reconstructies is jarenlang een belangrijk onderwerp van onderzoek geweest binnen het vakgebied van computer vision en image understanding als ook binnen de digitale fotogrammetrie. Er zijn verschillende methoden voorgesteld voor reconstructies op basis van luchtfoto's. Ondanks het vele onderzoek is er nog geen volledig systeem dat met voldoende betrouwbaarheid zelfstandig 3D reconstructies kan uitvoeren. Dit geldt met name voor complexe stedelijke gebieden waar zich gebouwen bevinden met een diversiteit aan verschijningsvormen. Uiteraard is er sprake van vooruitgang, maar er is nog veel vooruitgang te boeken. Deze vooruitgang kan bereikt worden door het combineren van meerdere databronnen en enige a priori informatie.

In dit project zijn grootschalige 2D GIS databases gebruikt als extra bron van informatie. Het combineren van luchtfoto's en kaartmateriaal bleek de betrouwbaarhied van de reconstructies te verhogen. Algemene kennis over de vorm van gebouwen is ook in het systeem geïntegreerd. Omdat de meeste gebouwen als een samenstelsel van eenvoudige gebouwentypes kunnen worden omschreven, kan de kennis omtrent het probleemgebied worden weergegeven in een bibliotheek die simpele modellen van gebouwen bevat. Daarom is een gebouwen-bibliotheek samengesteld uit de meest voorkomende oervormen, zoals een plat dak en verschillende vormen van zadeldaken.

Het reconstrucieproces voor gebouwen is geformuleerd als een hierarchisch schema voor het genereren en verificeren van hypotheses. Dit is geimplementeerd als een zoekboom. Er is een methode ontwikkeld die gebouwen lokaliseert in luchfoto's met behulp van kaartmateriaal. Daarnaast is een methode ontwikkeld voor het genereren van gebouwhypotheses die overeenkomen met de oervormen. Dit veronderstelt stereo matching van beeldkenmerken (hoeken, lijnen) die overeenkomen met kaartgegevens en het in overeenstemming brengen van gebouwhypotheses en beelden.

Een verdere bijdrage is het definieren van een maat om de gemaakte gebouwhypothese te evalueren om zo de hypothese te selecteren die het beeld het beste beschrijft. De maat is gebaseerd op het formuleren van wederzijdse informatie tussen het model van het gebouw en de beelden. Er is een analyse gemaakt van de methoden voor schatten van wederzijdse informatie uit oefen beelden. De maat is ontleend aan de informatie theorie en behoeft geen a priori informatie over de eigenschappen van het terrein of het object. De maat is robuust met betrekking tot schommelingen in de belichting.

Ook worden er geen veronderstellingen gedaan ten aanzien van de vorm van objecten. Dit betekent dat de methode vrij algemeen is en toegepast kan worden in een verscheidenheid aan applicaties. De geformuleerde aanpak voldoet aan de meeste voorwaarden voor een automisch systeem voor 3D reconstructies. Het ontwikkelde systeem is toegepast in stedelijke en sub-urbane gebieden om gebouwen te reconstrueren en heeft daarbij goede resultaten geboekt. De experimenten zijn uitgevoerd met twee data sets met verschillende karakteristieken. Het systeem was in staat om meer dan 80% van de gebouwen te reconstrueren. De precisie van de reconstructies bleek voldoende te zijn voor het maken van kaarten.

Ildikó Süveg, 2003

Acknowledgement

This research would not have been possible without the help and encouragement of many people. Naming all is not possible. Specifically, I wish to acknowledge the contribution of the following people to this thesis.

First, I would like to thank to my supervisor, prof. George Vosselman for believing in my capability to undertake this research project and for stimulating scientific discussions. This research would not be in its current shape without his continuous support.

Over the past four years, I have had the pleasure to work in the Photogrammetry and Remote Sensing group. My colleagues in the group created a pleasant and stimulating atmosphere. Also, in this period I have been fortunate in having a number of friends in Delft, who helped me keep a balance between work and social life.

Finally, my deepest gratitude goes to my mother, Anna. I dedicate the thesis to her.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Domain

3D reconstruction of buildings from aerial images has been an active research topic in Computer Vision as well as in Digital Photogrammetry in recent years. This can be explained by the fact that 3D reconstruction of buildings has become of increasing practical importance. Traditional application domains are those of cartography and photo-interpretation. In the cartography domain, a successful 3D reconstruction system would greatly reduce the effort needed to assemble a digital map product. Newer applications include urban planning, construction, environment, communication, transportation, energy and property management, tourism, and virtual tours of cities.

Photogrammetric methods are well established but show inefficiencies due to the extensive amount of data. Manual 3D processing of aerial images is very time consuming and requires highly qualified personnel and expensive instruments. Therefore, speeding up this process by automatic or semiautomatic procedures has become a necessity.

The current state of automation in the reconstruction of buildings from aerial images is still surprisingly low. A lot of algorithms and systems have been proposed towards this problem. However, a versatile solution to the automatic reconstruction has not been found yet, with only partial solutions and limited success in constrained environments being the state of art. The difficulty in obtaining a general solution to this problem can be attributed to the complexity of the reconstruction itself, as it involves processing at different levels: low level processing (feature extraction), middle level processing (representation and description of building models) and high level processing (matching and reasoning). The realization of an adequate reconstruction system depends upon the success at all these levels and in combining these levels of processing.

The reconstruction task is made difficult by several factors. The main factor is the complexity of aerial scenes. The built-up areas are very dense and there are many building types. Aerial images generally contain a great deal of information which is irrelevant with respect to the given task of building extraction, e.g. vegetation, cars, building details like windows, stairs etc. Hence, it is difficult to separate the useful information from irrelevant details. On the other hand there is a loss of relevant information due to the projection of three-dimensional shapes into two-dimensional images. Furthermore occlusions, low contrast or unfavorable perspectives will cause loss of information.

The current state of art in this field can be found in the proceedings of Ascona 2001 workshop [Baltsavias et al., 2001].

1.2 Goal of the Project

The goal of this project is to develop an automatic system for the 3D reconstruction of buildings from aerial imagery.

To handle the complexity of 3D building reconstruction, aerial image data is combined with other data sources. As additional information source, large-scale 2D GIS (Geographic Information System) databases are used. The GIS database contains outlines of the footprints of buildings among other information not important for the given task. This information is extracted from the GIS database and is used as a digital map. GIS databases are widely available for most well developed countries. By combining the images with GIS data, the specific strengths of both the images (high resolution, accuracy, and large information content) and the map (relatively simple interpretation) can be exploited.

Generic knowledge about the shape of the buildings is also incorporated in the system. Some of the most challenging tasks in 3D reconstruction of buildings occur in dense urban areas where scene clutter and variety of building types complicate the process. Therefore, a promising concept for automatic building reconstruction should incorporate a sufficiently complete model of the objects of interest and of their relationships.

The interior orientation of the camera and the exterior orientation of the images are assumed to be known or derived by use of some photogrammetric techniques. The exterior orientation of the cameras are assumed to be given in the same coordinate system as the map. Therefore the images can be considered registered to the map.

The requirements that the automatic 3D building reconstruction system has to fulfill relate to:

- robustness: An ideal system would be able to reconstruct all buildings of a scene. However, this is very difficult to achieve. Of course, the goal is to reconstruct as many buildings as possible, but at the same time the system should be able to indicate the buildings which cannot be reconstructed. We intend to describe the reconstructed scene in a "traffic light" manner. This means that the reconstructed building models will be classified in three categories:
  - green: buildings that are certainly reconstructed well.
  - yellow: buildings that are reconstructed, but the system is not sure about the result.
  - red: buildings that cannot be reconstructed.
- resolution: The resolution refers to the details of the reconstruction. This is highly application dependent. It also depends on the resolution of the aerial images. The level of detail of the reconstruction should match those of large-scale GIS maps. The main building and the attached shed have to be reconstructed, but small roof structures such as small dormers and chimneys can be omitted from the reconstruction.
- accuracy: The numerical accuracy of the reconstruction is important. It is highly correlated to the resolution and it is application dependent.
- computation time: Since the main objective of an automatic system is to reduce the amount of time required to create 3D models of scenes, the system should prevent excessive computation time. A fair computation time would be a number of minutes for one building structure. Regardless, the computation time is less important than the amount of the required corrections (yellow and red categories of buildings). So, a slower system with better performance is preferred to a fast system with lower performance.

1.3 Related Work

The problem of automatic 3D building reconstruction has been a central research topic in computer vision and image understanding communities and in digital photogrammetry for many years. A variety of approaches have been suggested for the reconstruction of buildings from aerial images.

The objective of this section is to briefly summarize existing approaches for the reconstruction of buildings from aerial images and to point out their strengths and weaknesses. To classify the different approaches, a few characteristics can be considered. These characteristics can be the user interaction (manual, semiautomatic, automatic systems), input data (monocular images, stereo images, aerial images and additional data sources), and building models (polyhedral models, parameterized models, generic models).

Since manual 3D processing of aerial images is time consuming, the development of automatic or, at least, semi-automatic techniques has become a necessity. In the semiautomatic systems automatic modules are integrated in an interactive work flow. The user needs to actively interfere to delineate and / or support the process of generating building hypothesis. The user often places the given object models in the image and these object models are then refined by the system. So, the system does the measurement task, whereas the user handles the interpretation and modelling tasks [Lang et al., 1995]. Although of practical importance, manual and semiautomatic techniques are not included in this review.

1.3.1 Monocular Image

Some of the early approaches attempted to work solely with monocular images. These systems exploit shadows either to infer the third dimension or to verify a generated hypothesis. Using only one image makes the problem more difficult as feature correspondence cannot be used to infer 3D and some ambiguities are harder to resolve in a single image. Consequently, often the buildings are assumed to be rectangular or rectilinear flat roofs.

The research group from IRIS (University of Southern California) uses perceptual grouping and shadow analysis for detecting buildings in aerial images [Mohan and Nevatia, 1989a], [Huertas and Nevatia, 1988], [Huertas et al., 1993]. Since buildings exhibit a great deal of geometric regularity, the features can be grouped based on the geometric relationships among them. The grouping criterion includes proximity, collinearity and symmetry. Lines, parallels, U-contours, and rectangles are identified. A feature hierarchy encodes the structural relationships specific to this set.

After the formation of all reasonable rectangles, a selection process is used to choose rectangles, which have strong evidences of support and have minimum conflict among them. The work of [Mohan and Nevatia, 1989b] used Constraint Satisfaction Networks for this purpose. Unfortunately, the network was not efficient and too time consuming.

Huertas proposed an improvement to this method [Huertas et al., 1993]. A local selection is performed at the beginning in order to reduce the number of groupings. Only the rectangles, which have enough local support, are retained for the subsequently global selection. In combination with the knowledge of the sun direction, regions can be verified by looking for shadow on the opposite side of the sun direction. Shadows and walls are used to help form and verify hypotheses generated by the grouping process.

The reconstructed buildings are relatively simple (flat, rectilinear) and the scenes are not very complicated. It is necessary that the line segments corresponding to the building edges be fully extracted even before the grouping process. The tests performed on suburban scenes showed a good detection of the buildings and low false alarms.

Another method for extracting planar polygonal rooftops in monocular aerial imagery was proposed by Jaynes [Jaynes et al., 1994]. Through bottom-up and top-down construction of perceptual groups, polygons in a single aerial image can be robustly extracted.

Orthogonal corners and lines are extracted and hierarchically related using perceptual grouping techniques. Features and their groupings are stored in a feature relation graph. Low-level features are nodes in the graph and binary relations between features are represented with an edge between the corresponding nodes. Both nodes and edges are assigned a certainty that reflects the confidence of a feature or a feature grouping. Cycles in the graph correspond to possible building roof hypotheses. If a cycle is not closed, the system searches locally in top-down manner for the missing feature. If evidence is found in the image, a virtual feature is hypothesized. Extraction of the best grouping of features into a building roof hypothesis is posed as a graph search problem. The maximally weighted, independent set of cycles in the graph is extracted as the final set of roof boundaries.

Irvin and McKeown [Irvin and McKeown, 1989] searches for L shaped shadow segments. They make the assumption that a concave corner belongs to a rectangular building. They use shadow to verify a hypothesized building similar to Huertas and Nevatia, but they also estimate the height of a building by measuring the length of the shadow.

Almost all of these methods make three key assumptions to constrain the interpretation of the image information: the image has nadir acquisition geometry, the buildings are rectangular shaped with flat roofs, and buildings are brighter than shadows. Given the first two assumptions, detection of right angle corners in the image becomes a key processing step, as these corners are the primitive features from which the structural hypotheses are constructed. If the acquisition geometry is oblique, then the right angle corners in space no longer correspond to right angle corners in image space. If the building structure is not rectilinear in nature, then right angle corners have little use as primitives for object extraction.

In [Shufelt and McKeown, 1993] and [McGlone and Shufelt, 1994], the use of vanishing point information as constraint on the construction of primitives from intermediate features reduces the combinatorics of hypothesis search. The use of projective geometry in conjunction with a camera model provides the ability to analyze shadow shape and object photometry for hypothesis generation (shadow based extrusion) and verification (shadow based and illumination based consistency testing of building models).

The methods, which work with single images, are mainly based on the grouping of line segments or corners. However, the results of the low level algorithms for feature extraction are limited.

1.3.2 Stereo Images

Traditional photogrammetric techniques always use stereo image pairs to do measurements. There are a lot of systems that use widely available stereo images. Stereo images allow the determination of the third dimension by epipolar matching of different features extracted from both images. Multi-view strategies are advantageous in providing redundant information and improving the accuracy of the reconstruction.

These systems can be classified in two categories: systems that are extensions of monocular systems for working with stereo images and systems that were designed from the beginning to work with stereo images.

The systems from the first category actually process the images separately and the stereo images are used to validate the hypotheses generated by analyzing the images separately. Some of these systems are described briefly below.

In [Noronha and Nevatia, 2001] a system is described that detects and constructs 3D models for rectilinear buildings with either flat or gable roofs from multiple aerial images. This system is the extension of the monocular building detection system developed at IRIS. Hypotheses for rectangular roof components are generated by grouping image lines in each view. The hypothesis generation process combines tasks of hierarchical grouping with matching at successive stages. The hypotheses are verified by searching for presence of predicted walls and shadows. The approach makes extensive use of monocular analysis, even though multiple images are available.

Recently Jaynes [Jaynes et al., 1996] extended his system with stereo images. If an unclosed U contour hypothesis is found, a digital elevation model (DEM) is used to select a close surface. In addition, three roof types are used for checking the correctness of the detected surface starting from a 2D polygon and a DEM. A multi-view analysis is used to correct the errors of the method.

The system described in [Roux and McKeown, 1994] is based on the successive incorporation of new image data into an existing partial solution. This system is actually a further development of the monocular system of Irvin [Irvin and McKeown, 1989]. The building hypotheses are generated starting from hypothesized corners extracted from multiple views of the scene. This approach allows reconstruction of primitives not seen in the first images and increases the positional accuracy of reconstructed primitives by simultaneous solution of the collinearity equations. The system can work with oblique views.

The second category of systems actively uses stereo images. They aim to derive 3D information in the early stages of the processing. In this way object modelling can be done in 3D right from the beginning.

The Ascender I system was developed at UMASS (University of Massachusetts)for building detection and reconstruction from multiple aerial images of a site [Collins et al., 1995]. The Ascender system hypothesizes potential buildings in an image, automatically locates supporting geometric evidence in other images, and determines the precise shape and position of the building by multi-image triangulation. Ascender I was further developed and resulted in Ascender II [Hanson et al., 2001], which focuses on the use of multiple alternative reconstruction strategies from which the most appropriate strategies are selected by the system based on the current processing context.

The approach presented in [Bignone et al., 1996] relies on hierarchical hypothesis generation. It extracts 2D lines from a source image and computes their photogrammetric and chromatic attributes and their geometric relationships. Using geometry and photometry the 3D location of these edges are computed. The 3D segments are then grouped into planes and 2D enclosures are extracted and combined with the 3D planes to form 3D patches. Finally, these 3D patches are ranked according to their geometric quality and the best are retained for the candidate 3D object models.

The IMage Processing for Automatic Cartographic Tools (IMPACT) project is a five site collaboration funded under the European Community Esprit Long Term Research Programme. The partners are the Universities of Bonn (Germany), Leuven (Belgium), Oxford (England), Ecole Nationale Superieure des Telecommunications (ENST, France), and Eurosense (Belgium). The input data used in this project are high resolution color images. At least three images of the scene taken from different positions are available.

Within this project, Bonn focused on generating a symbolic 3D description of the scene from symbolic 2D image descriptions. The 2D description consists of a Feature Adjacency Graph (FAG), for example a 2D corner and its associated line segments and regions. The idea is to transfer the neighborhood relations from the images into object space and to use them to produce consistent 3D object descriptions. This involves matching the 2D descriptions over multiple images. They propose a model-based approach for 3D extraction of buildings.

This approach relies on the well-defined combination of the building part models [Fischer et al., 1997], [Fischer et al., 1998].

The approach developed at Leuven constructs a polyhedral model of the roof structure that captures the topology of the roof. First, regions are selected and afterwards 3D line segments are generated by matching line segments belonging to the same regions. These 3D lines are grouped and polygonal patches are formed. In a next stage the different polygons are glued together into a roof model. To improve the metric accuracy this roof model is fit to the image data [Moons et al., 1998].

Oxford developed an algorithm for automatically matching line segments over multiple images. The algorithm employs geometric constraints based on the multi-view geometry together with photometric constraints derived from the line neighborhood [Schmid and Zisserman, 1997]. The second development is a method for automatically computing a piecewise planar reconstruction based on the matched lines. The novelty here is that a planar facet hypothesis can be generated from a single 3D line, using an inter-image homography applied to the line neighborhood [Baillard and Zisserman, 2000].

ENST employs a planar approximation on a region of the scene, starting from a relatively sparse disparity map. The regions are delineated by a color segmentation algorithm. A plane is computed using the 3D data available for the region and afterwards the equation of the surface is refined by statistical tests [Girard et al., 1998], [Fradkin et al., 1999].

1.3.3 Multiple Data Sources

The reconstruction of buildings using only aerial images as data source has been proven to be a very difficult problem. The complexity of the reconstruction can be greatly reduced by combining the aerial images with other data sources. These data sources can be Digital Surface Models (DSM) and / or scanned or digital maps, which are largely available for many countries. DSM can be derived by stereo matching from aerial images or measured directly by laser scanner systems.

The maps describe the ground plans of the buildings. By combining the images with maps, the specific strengths of both the images (high resolution, accuracy, large information content) and the map (relatively simple interpretation) can be exploited. First of all, the maps provide useful information about the shape of the buildings. Also, the maps can provide focusing areas, thus feature extraction, matching or grouping can be done on smaller areas. Consequently the combinatorics of the reconstruction process is reduced.

One can distinguish two classes of approaches for integration of multiple data sources [Vosselman and Suveg, 2001]. The differences between these two classes are given by the way in which building models are generated. One class of approaches is model driven, where building models are defined using only the information available in the map and then verified in the images [Pasko and Gruber, 1996], [Haala and Anders, 1996]. The other class of approaches is data driven [Jibrini et al., 2000]. Building models are generated by deriving 3D information from the stereo images and the information contained in the map is used to constrain the possible building hypotheses.

In [Pasko and Gruber, 1996] a method based on fusing aerial image data, 2D GIS maps and Digital Terrain Model (DTM) data is presented. The ground plans of the buildings and the lines extracted from an image are related by an affine matching procedure. This matching procedure requires approximate knowledge of the height of the buildings and approximate elevation data of the ground. The results of the matching are improved and verified by using a second aerial image.

In [Haala and Anders, 1996] the map assures detection and localization of the buildings in the images. The information about the shapes of the buildings available in the GIS map is used to divide the buildings into rectangles. For each rectangle multiple parametric building models are defined. These models are verified and the unknown parameters are determined by matching the extracted lines from the images against the lines of the building models. In this way the buildings are described as combinations of building boxes.

A data driven approach is presented in [Jibrini et al., 2000]. First, the modelling of the inner surface of the building roof by planar hypotheses extracted by a 3D Hough transform on a fuzzy DSM is carried out. The DSM is computed inside the ground footprint given by the cadastral map by matching windows with contour adaptive shapes. Afterwards the 3D outer limits of the roof are recovered by repositioning the cadastral 2D segments in 3D using the previous planar hypotheses and the image contrasts. The map information was employed in all the processing steps to reduce the space of admissible solutions.

1.3.4 Discussion

The work done on 3D scene reconstruction reveals a variety of modelling schemes. The diversification of data sources is the turning point in the evolution of 3D building reconstruction systems in the last few years. While the first 3D reconstruction systems from the late 80's used only monocular images, the new trend consists of multi-view and even multi-data approaches. The usage of multiple data sources allows reduction of ambiguities by accumulation, evidence reinforcement or complementarity. However, the integration of different data sources is difficult and can raise other problems. For instance, the number of primitives obtained by grouping increases in a multi-view approach. Hence, the number of hypotheses increases. This can lead to a combinatorial problem.

Most of the approaches proposed for 3D building reconstruction are based on the extraction of image features (line segments, corners) followed by the grouping of these features. However, low-level methods typically fail to extract all relevant features and often find spurious ones. This difficulty is handled by using very constrained models, such as flat rectilinear roofs. These models impose very severe restrictions and are not usable in many areas.

Another solution for the problem of the 3D building reconstruction can be the combination of 2D and 3D processing. The cooperation between 2D and 3D processing can be carried out at different levels:

- The images are processed separately and afterwards the results are fused. The depth maps are exploited after the detection and grouping of primitives in 2D. The depth maps are used for validating the hypotheses generated by monocular analysis.
- First a depth map is computed. This map is used already during the grouping of primitives.

Another major evolution is the resolution of the images. Most of the work has been done on aerial images with a resolution better than 30 cm/pixel. However, the resolution is highly dependent on the application. For an industrial site where buildings are dispersed a resolution in the order of meters is sufficient. But for a dense urban site a higher resolution is required. The resolution of images used in the IMPACT project is very high (10 cm/pixel). Most systems work with gray scale images, and only a few systems work with color images.

Despite considerable research effort, there is no complete system that can reliably perform autonomous 3D building reconstruction in a wide variety of scene domains. This is particularly true in complex urban areas containing buildings with different shapes and roof types as well as in complicated underlying terrain. Of course, some progress has been made, but there is room for improvement. We believe, this improvement can be achieved by fusing multiple data sources and some a priori information.

1.4 Overview of the Method

Our approach relies on combining pairs of stereo images (Figure 1.1) with 2D GIS (Geographic Information System) databases and domain knowledge.

In the first stage the buildings are localized in the images based on the information from the ground plans of the buildings contained in the GIS database. This

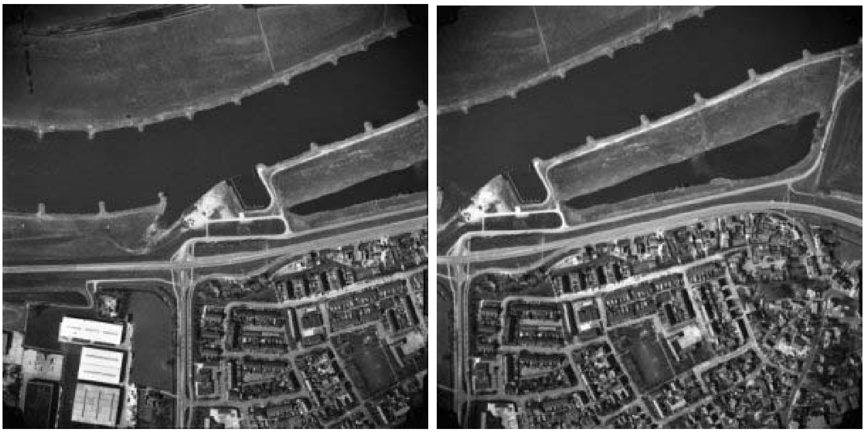


Figure 1.1: Pair of stereo images

restricts the processing at all succeeding stages to one building structure. The fact that the reconstruction process is focused on one building, greatly reduces the complexity of the reconstruction.

To cope with the complexity of aerial images, specific knowledge about buildings is integrated in the reconstruction process. Since most buildings can be described as an aggregation of simple building types, the knowledge about the problem domain can be represented in a building library containing simple building models (flat roof, gable roof, and hip roof building). The approach of modelling buildings using a set of basic building primitives suggests the usage of Constructive Solid Geometry (CSG) representation for building description. In this way, a complex building can be seen as a CSG tree, where the leaf nodes contain primitive building models and the internal nodes contain boolean operations such as union, intersection, and difference.

The building reconstruction process is formulated as a multi-level hypothesis generation and verification scheme and it is implemented as a search tree (Figure 1.2). The tree is generated incrementally by the search method.

The first step of the actual reconstruction process is the partitioning of a building into simple building parts, which might correspond to the building models defined in the building library. First, the partitioning is performed using only the ground plan of the building defined in the digital map. If the ground plan of the building is not a rectangle, then it is assumed that it can be divided into rectangles, called partitions. Then, a partitioning scheme can be defined as a subdivision of a building into disjoint partitions. Usually, a building can have multiple partitioning schemes.

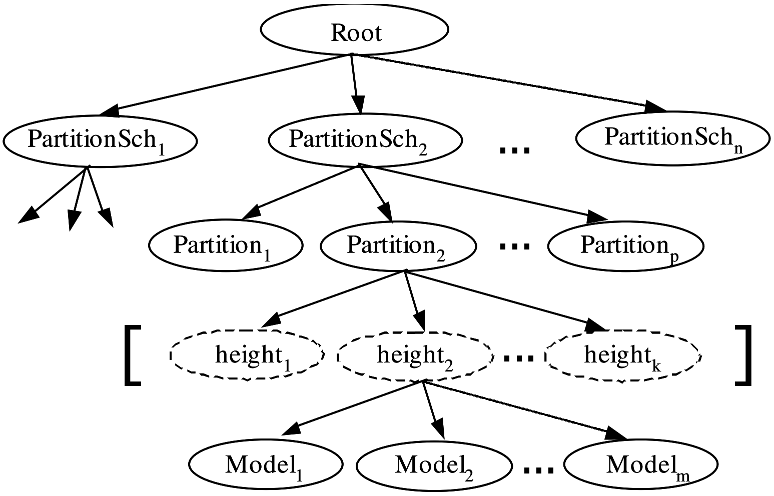


Figure 1.2: Search tree

All the possible partitioning schemes of a building are represented on the first level of the search tree. To avoid a blind search method of the tree, MDL (Minimum Description Length) principle is used to rank the partitioning schemes. This principle provides a means of giving higher priority to the partitioning schemes with a smaller number of partitions. The second level of the tree contains the partitions corresponding to each partitioning scheme.

If the terrain is not flat, then the third level of the tree contains the possible height values that have to be checked for each partition. Next, the tree is expanded with a level corresponding to the different building hypotheses generated for each building partition. Corresponding to each building primitive defined in the building library, a building hypothesis is generated. The building hypotheses are refined by fitting them to image data. Afterwards, the hypotheses are verified by back projecting them into the images and matching them with the information extracted from the images. The matching defines a score function that allows comparison and evaluation of different building hypotheses. So, this function can be used to guide the search in the tree. The score function is based on the formulation of the mutual information between the building model and the images.

The CSG tree representing a building is given by the best fit of the building models corresponding to the building partitions. In the final verification step the complete CSG tree is fitted to the image data. To improve the results, constraints, which describe geometric relationships between building primitives, are incorporated in the fitting algorithm.

1.5 Outline of the Thesis

The second chapter presents the domain knowledge integrated in the our building reconstruction method and the usage of this knowledge to localize the buildings into images and to partition a complex building into building primitives. The third chapter describes the generation of building hypotheses by fitting the building primitives from the building library to image data. Chapter 4 presents the metric used to guide the search in the search tree. It contains an overview of the information theory necessary to understand this metric. It also discusses the estimation of the mutual information from samples. The fifth chapter contains a wide variety of building reconstruction experiments designed to validate our approach. The final chapter includes conclusions and directions of future research.

The work in this thesis is based on some papers published over the course of the research [Suveg and Vosselman, 2000, Suveg and Vosselman, 2001, Suveg and Vosselman, 2002a, Suveg and Vosselman, 2002b].

Chapter 2

Knowledge Based Preprocessing

In this chapter the domain knowledge integrated in our building reconstruction system is presented. This knowledge includes the GIS database describing the ground plans of the buildings and the building library containing building primitives. Depending on the processing context a building can be represented in different ways. These different representation types for describing building models are discussed in the first part of the chapter.

The second part of the chapter describes the localization of the buildings into the images and the partitioning of the buildings into simple building-parts. Both of these processes make use of the domain knowledge described in the first part of the chapter.

2.1 Domain Knowledge

2.1.1 2D GIS Database

Our approach makes use of additional information sources of the scene besides aerial images. This information can be available either from different kinds of maps or from Geographic Information System (GIS) databases. For the work presented here we use information extracted from a GIS database. Actually, we are only interested in the layer of the GIS database that contains the footprints of the buildings (Figure 2.1). This observation allows us to simplify the representation

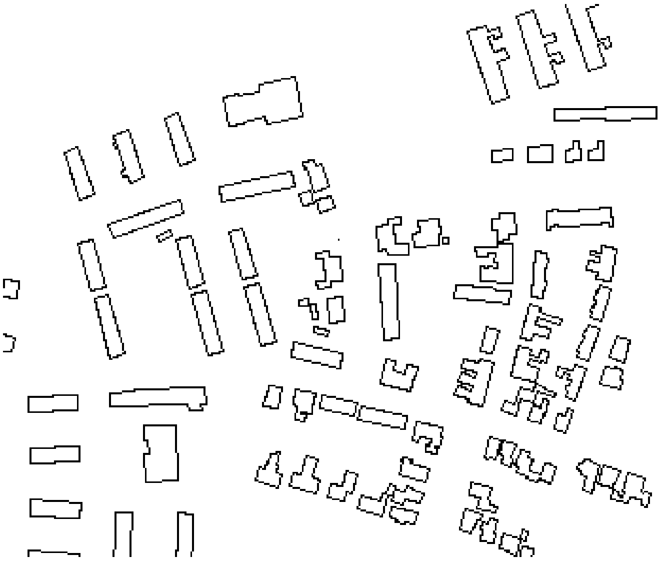


Figure 2.1: Footprints of the buildings extracted from a GIS database

of the map knowledge by merely extracting this layer. The map can be described in BNF as:

```
<map> ::= <building> | <map> <building>
<building> ::= <number> <footprint>
<footprint> ::= <corner> | <footprint> <corner>
<corner> ::= <number> <coord X> <coord Y>
```

Clearly, the 3D reconstruction of buildings can take advantage of the 2D maps that contain the ground plans of the buildings. However, when used for image analysis, map information should be considered imprecise and uncertain, so that it only provides a rough model of the scene. Our goal is to extract different cues useful for the analysis of the aerial images. So we are indeed interested in processing of the aerial images that are more accurate and make use of map data to help the image interpretation.

As we shall see later on, the footprints of the buildings will be used in almost all steps of our approach. First, a building can be localized in an image and its region of interest can be delineated in the image. Then the rest of the processing is done only on this region. Secondly, the footprint can give a good hint about the structure of the building. Therefore, it can be used to derive hypotheses on the decomposition of the building in simple building primitives. Also, the footprint can provide the initial values at the generation of building hypotheses.

2.1.2 Building Models

A building reconstruction system needs an internal model of the objects (buildings) to be acquired. Buildings reveal a high diversity in structure, but at the same time, most buildings show regularities. These regularities represent the domain knowledge which is used in all stages of the reconstruction. The domain knowledge can be represented implicitly by a set of rules applied in the reconstruction process, or explicitly by providing a database of explicit building models.

Shape representations for 3D buildings are discussed in terms of their primitive elements that are combined to form the solid bodies. For a given representation, the coordinate system may be viewer-centered or object-centered. In the first case, locations in 3D space are specified with respect to an origin at the viewing position, and in the second case with respect to a reference point and axes of the body.

Although there are several representations, which are common for buildings, we shall concentrate here on three common types of primitive based models. These are volumetric, surface (B-rep) and wire-frame models. Depending on the processing context a building can be represented in one of these ways.

Wire-frame representation describes the object as vertices and edges, but not surfaces or faces. Since many early computer vision systems worked with polygonal objects, edges have been the main local feature used for recognition. The wire-frame representation assumes that the surfaces of the object are planar and that the object has only straight edges. It is used to project the objects into the image.

Boundary representation (B-rep) describes the object in terms of its boundary elements. An object has three sets: its edges, its vertices and its faces. Thus, the representation is a directed graph containing face, edge and vertex nodes. A vertex has an associated 3D point and a set of edges that meet at that point. An edge has a start point, an end point, a face to its left, a face to its right and an arc that defines its form. A face has a surface that defines its shape and a set of boundaries, including its outer boundaries and its holes. A boundary has an associated face and a set of edges. Vertices constitute the basic geometric information, while edges and faces are defined through topological relations. This type of representation is frequently used for visualization purposes since the visualization algorithms and this representation are well suited.

Volumetric representation: Volumetric models are the most intuitive for describing solids. CSG (Constructive Solid Geometry) is well suited to describing

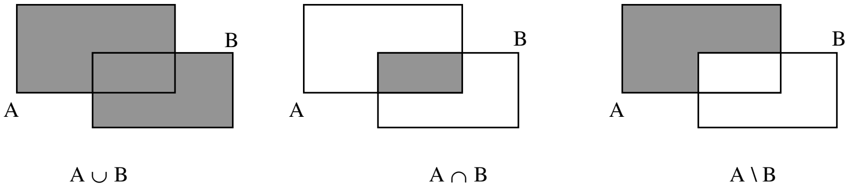


Figure 2.2: CSG Boolean operations

complex shapes, which can be composed from a set of primitives. Within this approach, buildings are described by combining a set of basic primitives. The primitives are combined by means of the boolean operations union, intersection, and difference (Figure 2.2). Finally, a building will be described as a CSG tree, where the leaves of the tree contain primitives and the internal nodes contain boolean operations.

The choice of the representation of primitives is essential for building reconstruction. Referring to different degrees of freedom within the models, one can distinguish three types of representation:

- *Specific models*: They have fixed topology and fixed geometry. They are mainly used in CAD applications, where a precise model of the object is known. This is seldom case of buildings.
- *Parametric models*: They have fixed topology and variable geometry. Many simple buildings can be described using a few parameters. Therefore, an approach to reconstruct buildings can use parameterized models describing the most common building types. These building types can be described in a building library. However urban scenes show a large variety of building types, and they require a well-populated building library. Nevertheless, this is difficult to acquire. An alternative could be a building database containing building parts, since a variety of building models can be generated from a relatively small class of predefined building parts.
- *Generic polyhedral models* are more general and describe objects without a fixed structure but using only topological aspects (Figure 2.3).

The modelling process results in a CSG tree, whose interior nodes contain operations and the leaves contain instantiated primitives (their pose and shape attributes are known). Figure 2.4b shows the CSG tree describing the building from Figure 2.4a.

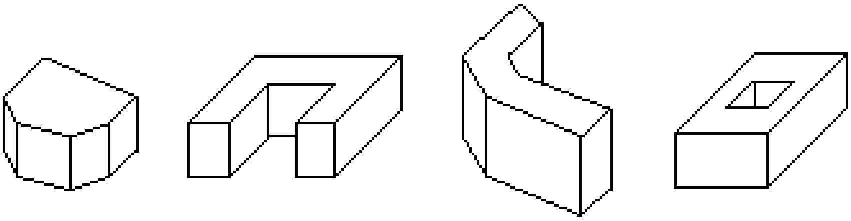


Figure 2.3: Generic polyhedral building models

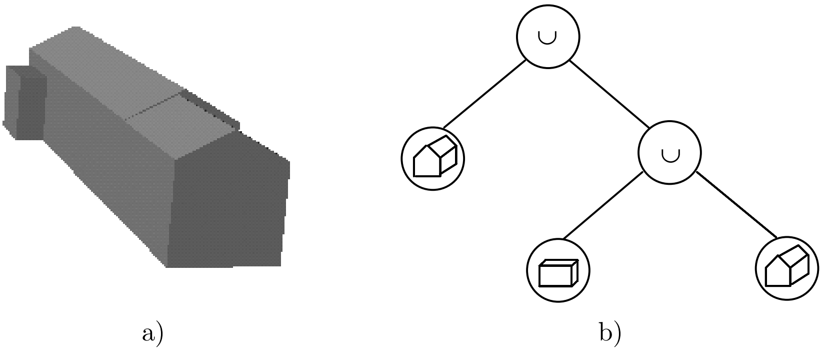


Figure 2.4: a) 3D model of a building and b) corresponding CSG tree

2.1.3 Building Library

A particular choice of representation should be well suited for the problem domain. Systems, which are meant for natural scenes, face entirely different representational problems than those for man-made environments where the objects are regular, well defined shapes. Another important aspect of this choice is its overall representational power. The chosen set of primitives ought to combine to span most of the objects in the domain. The set of primitives should be just large enough that their combinations represent most of the object shapes in the domain of interest, but not so large to make acquiring of them difficult. Also, the choice of representation should be based on the degree to which the models can be extracted from the available data.

Most buildings can be described as aggregation of simple building types. Starting from this observation, the knowledge about the problem domain can be represented in a building library containing the simple building primitives. As basic building primitives, we can consider a flat roof, a gable roof and a hip roof building, resulting

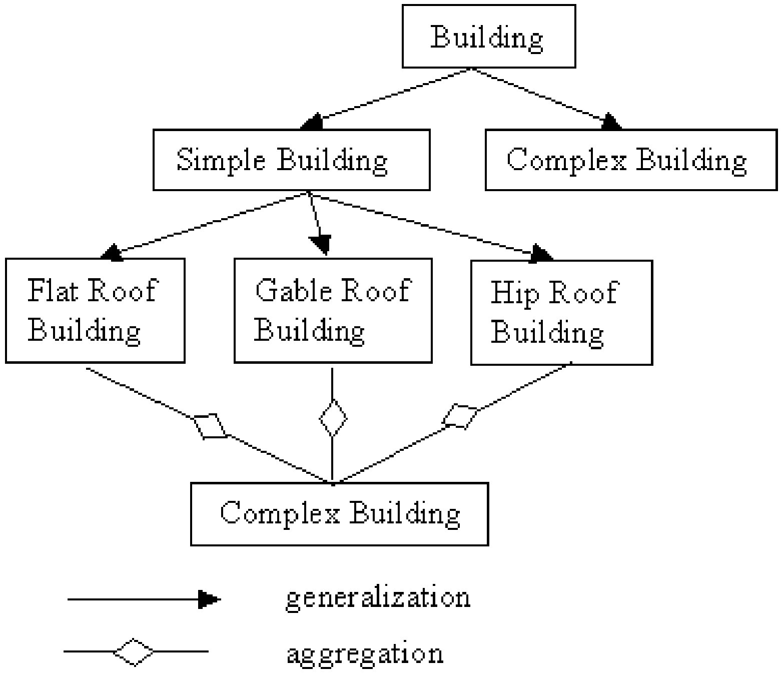


Figure 2.5: Building hierarchy

in the building hierarchy shown in Figure 2.5. The choice of these primitives satisfies all of the above-mentioned goals. They are well suited for modelling a variety of buildings that appear in aerial images and they combine to represent a large number of building structures.

The approach of modelling buildings using a set of basic building models (primitives) suggests the usage of CSG representation for building description. The basic building primitives of the CSG representation can be described by parametric models.

Parametric models control the deformations using a set of parameters that are capable of encoding a specific characteristic shape and its variations. Different model class instances can be obtained using different parameter values. This type of model is commonly used when some prior information of the geometrical shape is available, which can be encoded using preferably, a small number of parameters.

Parametric building models can be described by shape and pose parameters. The shape parameters describe the geometry of the building primitive while the pose

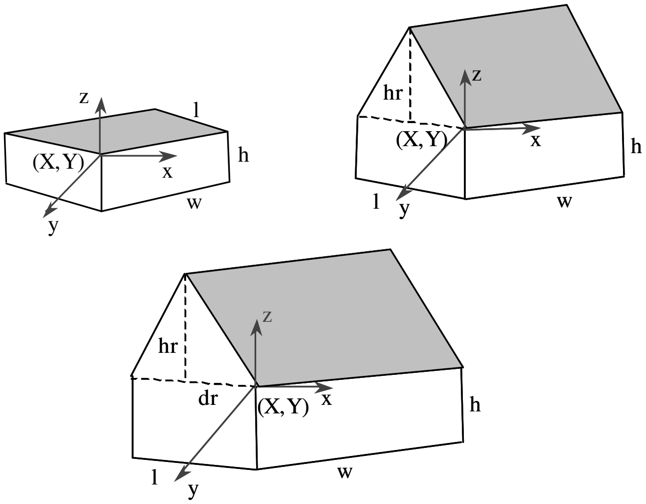


Figure 2.6: Parametric building models a) Flat roof building b) Symmetrical gable roof building c) Non-symmetrical gable roof building

parameters represent the relation of the building primitive with respect to the world coordinate system. The a priori knowledge about the structural properties of a class of buildings is encoded in the model.

The first of the primitives to be used throughout this work is a flat roof building (Figure 2.6a). A rectangular volume encodes the geometrical properties of this type of building. To describe a flat roof building primitive 6 parameters are necessary: 2 shape parameters and 4 pose parameters. The shape parameters are: width (w) and length (l). The pose parameters are: x, y, z coordinates of the buildings' reference point and the orientation in the xy-plane. Note that the z coordinate is composed from the height (h) of the building primitive and from the height of the terrain.

Another primitive is a gable roof building composed from a rectangular volume and a triangular volume. Two types of gable roofs are used: symmetrical and non-symmetrical gable roof primitive. For a symmetrical gable roof primitive (Figure 2.6b) an extra parameter, the height of the ridge (hr) has to be added to the parameters of a flat roof primitive. For a non-symmetrical gable roof primitive (Figure 2.6c) two extra parameters, the height of the ridge (hr) and the distance from the roof reference point to the ridge base point (dr) has to be added to the parameters of a flat roof primitive.

The building models implicitly contain all the relations and constraints among edges and faces necessary to specify the complete polyhedral object.

2.2 Localization of Building Primitives in Aerial Images

Most of the building reconstruction strategies have two main parts: the localization step and the actual reconstruction step. The localization of the building in the images means the detection of regions of interest where the buildings lie. By first localizing the buildings in the images the reconstruction process can be focused on one building structure. This reduces the complexity of the reconstruction by a large amount. Therefore, it is desirable to localize the buildings in the images first and afterwards to do the actual reconstruction.

In this section the determination of the region of interest for buildings and wall primitives is described.

2.2.1 Localization Methods

Four different approaches for detecting buildings can be distinguished depending on the data sources.

In some approaches the detection is done interactively by a human operator. The operator marks the regions of interest in the images [Henricsson, 1996].

Other approaches try to detect the buildings without additional data sources using only aerial images. This is more difficult because buildings are not explicitly represented in images. In this case the two steps, localization and reconstruction cannot be clearly distinguished. The localization step can make use of color, boundary appearance or implicit geometric model knowledge. Thus the localization step performs some tasks which could more or less belong to the reconstruction step. For instance [Jaynes et al., 1997] searches for rectangular structures in orthophotos and assumes that these rectangles are the boundaries of the buildings.

3D data sources such as DSM or DEM can also be used. It is easier to separate the relevant information for the given task from the rest of data. Haala [Haala, 1994] uses a DEM to detect buildings. He does local height thresholding to segment the DEM and computes the size and the compactness of the regions. Based on these properties he chooses the regions, which have building like attributes.

Recently, scanned maps [Maitre et al., 1995], or digital maps [Haala and Anders, 1997], [Pasko and Gruber, 1996], [Axelsson, 1998] have been used to focus the reconstruction. These map data provide quite accurate information for building detection. The ground plans of the buildings available in the map are projected into the images and the reconstruction is started from the interior points. The disadvantage of this approach is that buildings that are not contained in the map cannot be detected and thus reconstructed.

Our approach for locating buildings in aerial images is a hybrid one. It also makes use of the ground plans of the buildings contained in the digital maps. Two cases are distinguished. In case of terrain with small height variations (less than 15 m) the buildings can be localized in the images by using only information provided by the map. Applying the same method in terrain with larger height variations would result in regions too large for building locations. Hence the complexity of the reconstruction process would not be reduced. Therefore, an extension of this method for reducing these regions was required. This new method employs some higher level image understanding processes. In this case the building localization can no longer be separated from the building reconstruction process anymore.

2.2.2 Region of Interest for Buildings

Information about the ground plan of the building contained in the GIS data can be used to delineate a building in an image.

The map used is a two-dimensional digital map representing the ground plans of buildings. The buildings in the map are described by the list of the coordinates of their corner points. This knowledge source should be as robust as possible but cannot be expected to be error free. The errors / uncertainties can influence the reconstruction process. In the building localization process the uncertainties are due to: the unknown height of the buildings, the accuracy of the map, the roof extensions, and the feature extraction.

In order to handle these uncertainties we designed a two-step method. In the first step the uncertainty due to the unknown height of the buildings is handled, by assuming the height of a building to be between two extreme values. By projecting the ground plan of the building into the image for each of these values two contours are obtained. These contours have to be concatenated in order to get the area where the building is located. In the next step the contour obtained after the concatenation process has to be dilated for taking into account the other uncertainties mentioned above.

Concatenation of the contours

If the GIS data contained 3D information, i.e. height information, then it would be easy to find the exact position of a building in an image. The building could be

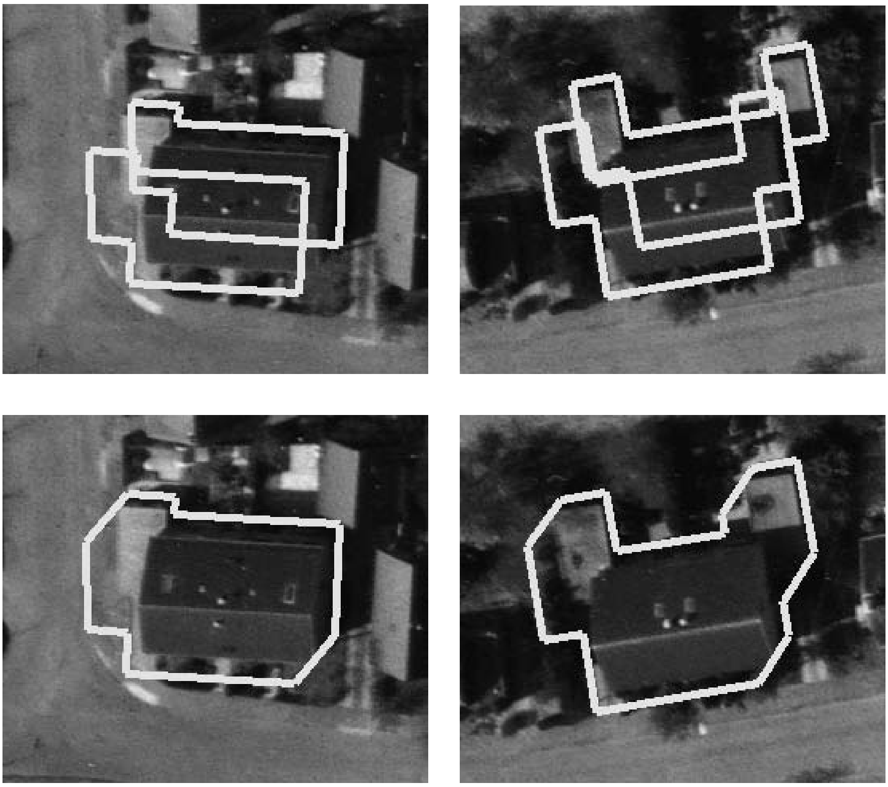


Figure 2.7: Concatenation of building contours. Top projected contours. Bottom concatenated contour

projected into the image using the projection equation with the known orientation parameters of the camera (see equation A.7 from Appendix A.1). However, the map contains only 2D information. Therefore an assumption about the missing third dimension has to be made. The minimum and maximum possible heights for the buildings from a data set can be hypothesized. For instance, in the area of our test data the height of the buildings could be assumed to be between 3m and 15m above ground level.

For these extreme height values, the ground plan of the building contained in the GIS data can be projected into the image using the known orientation parameters. In this way two slightly shifted contours are obtained in the image (Figure 2.7 top). Supposing that the real height of a building is between the two chosen extreme values, the position of the building in the image will be in an intermediate position between these two contours. Therefore, in the next step, these two contours have to be concatenated in order to get the maximal possible area in which the building might be found (Figure 2.7 bottom).

Dilation of the Contour

The resulting contour would represent the location of the building if all knowledge sources were error free. However, we also have to handle the uncertainties due to:

1. Accuracy of map data - Among the uncertainties of a map data, the positional accuracy of a map is important. In fact, positional accuracy of GIS data is only a minor source of uncertainty. The uncertainty introduced by the map, described by the standard deviation, is 20 cm, which corresponds to about 2 pixels in image space ($\sigma_1 = 2$).
2. Roof extension - Usually a map contains the ground plan of a building, which is smaller than the roof base of a building, because of the overhang. On the other hand, in an image the roof base can be seen. The value of this difference could be 50 cm, corresponding to about 5 pixels, which results in a mean of 2.5 pixels ($\mu_2 = 2.5$) and a standard deviation of 1.25 ($\sigma_2 = 1.25$) considering a 95% confidence interval.
3. Feature extraction - The features are extracted using the Förstner operator [Foerstner, 1994], which can be used for interest point extraction as well as straight line extraction. This operator assures a subpixel accuracy of the position of the extracted points and lines. Hence σ_3 is at most 1. However, the quality of the extracted features can be influenced by the acquisition of the image (high noise). This can affect the number, the shape, and the position of the extracted features.

To deal with these uncertainties the contour obtained after the concatenation has to be dilated. The value used for dilation can be computed from the uncertainties introduced by the three error sources mentioned above. The total standard deviation can be obtained using the formula:

$$\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_2^2} \tag{2.1}$$

Considering a 95% confidence interval, the interval is $[\sigma - 2\mu, \sigma + 2\mu]$. In our case the dilation parameter will be 9 pixels.

This dilation operation can be related to the concept of dilation from mathematical morphology. In this context the dilation can be defined as:

$$D(A, S) = A \oplus S = \bigcup_{s \in S} (A + s) \tag{2.2}$$

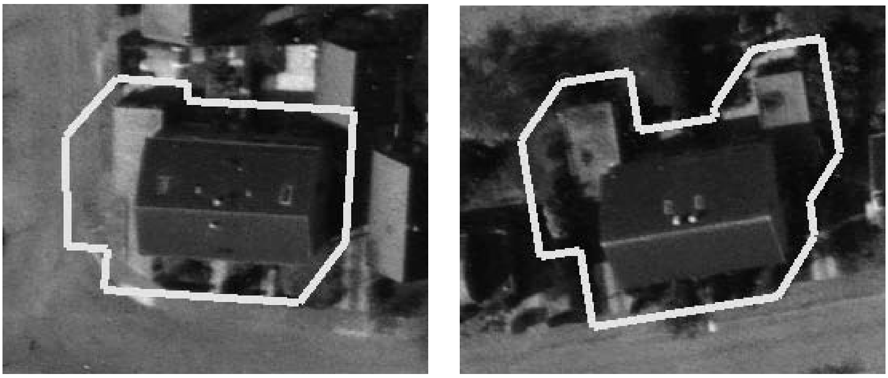


Figure 2.8: Dilation of building contours

where A is the object set, S is the structuring element and $\oplus$ denotes the Minkowski addition.

The amount and the way that the objects dilate depend upon the choice of the structuring element. For our application a squared structuring element is appropriate. The size of the square is given by the total standard deviation of the uncertainties. This structuring element is applied perpendicularly to the edges of the contour. After the dilation a new contour is obtained inside of which lies the building of interest (Figure 2.8). Figure 2.9 shows the delineated buildings in a scene.

2.2.3 Region of Interest for Wall Primitives

Based on the information contained in the map, not only a building but also the wall primitives, namely corners and segments, can be localized in the images. Actually, the localization of wall primitives is a more constrained case of a building localization process. The same uncertainties have to be considered but dealing with them is much easier.

For a wall corner point two points are obtained by projecting a corner from the map into the image for the considered extreme height values. The concatenation process is thus reduced to the connection of the two points by a segment. Next, dilating a segment means building a rectangle around the segment (Figure 2.10). Also, for a wall segment, instead of a polygonal contour, we have to work with a trapezoidal form. (Figure 2.11). After the locations of the wall primitives are found the image features, which may correspond to these wall primitives, have to be determined.

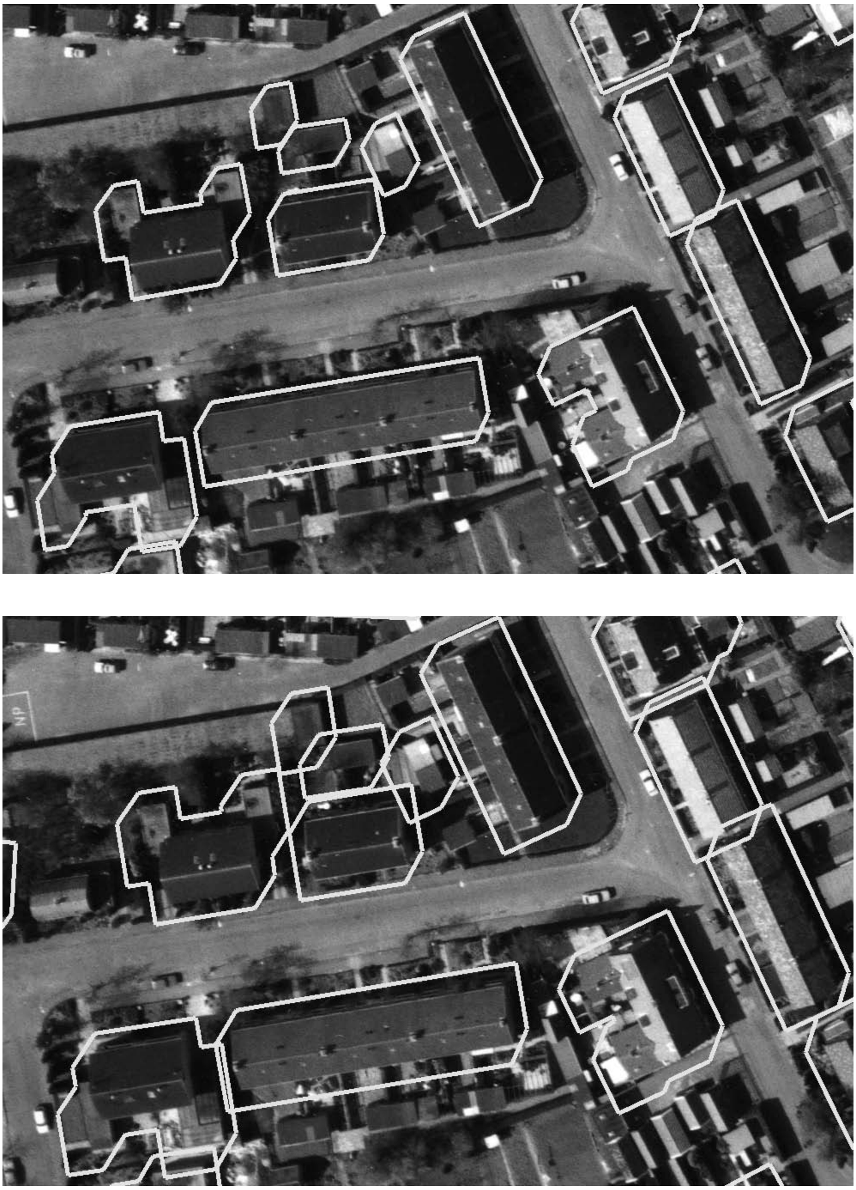


Figure 2.9: Localization of the buildings. Top contours after concatenation. Bottom contours after dilation

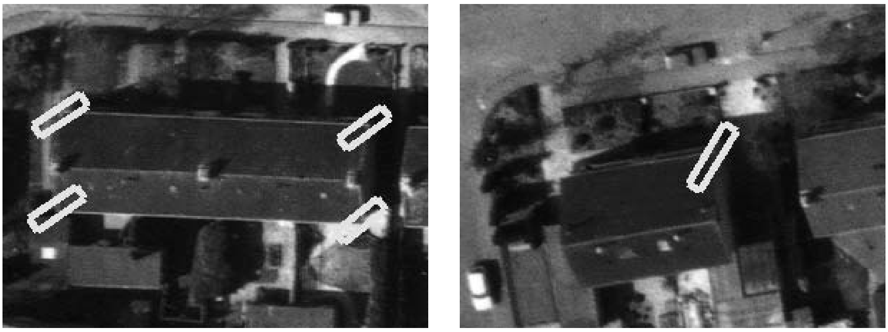


Figure 2.10: Localization of wall corners

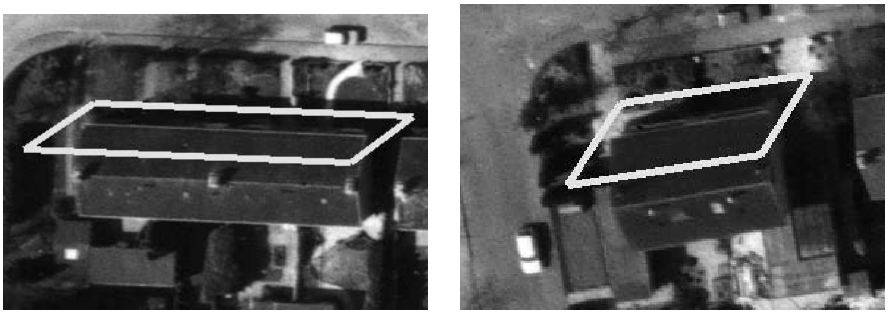


Figure 2.11: Localization of wall segments

The image features that are used as primitives for the building reconstruction are interest points and lines. Using this localization procedure image features can be classified as wall features or roof features. The image features, which are inside of a region defining the location of a wall primitive, might correspond to the wall primitive. Hence, they are labelled as wall features. The other features which lie inside of the region of interest of the building and do not correspond to any wall primitive are labelled as roof features.

2.2.4 Localization of Buildings

The method described in the previous section works well in case of small height variations of the terrain, i.e. close extreme values. The method can handle variations of up to 15 m in height, depending on the focal length. The region defining the location of a building is small, including only the given building structure.

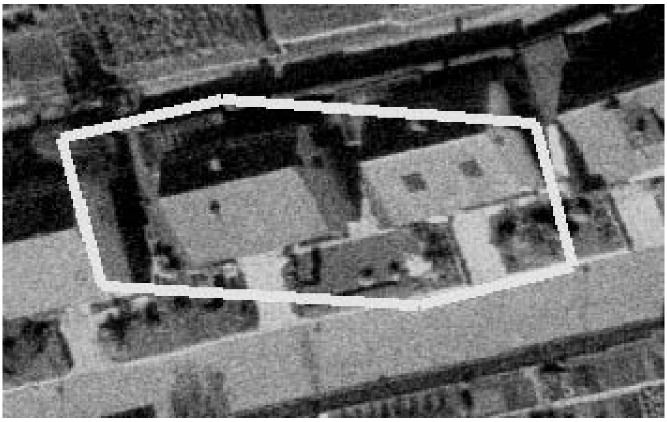


Figure 2.12: Localization of a building in case of large height variations of the terrain

However, applying the method for larger height variations would increase the obtained region where the building lies. In case of a very oblique view this region can become so large that it may include multiple building structures (Figure 2.12).

We have extended this method to be able to handle large variations in height. Our algorithm reduces a large interval of height values to a few individual height values which have to be checked afterwards by some higher level image understanding methods. Therefore the building localization can no longer be separated from the building reconstruction process.

In the case of large height variations, the height uncertainty we have to deal with includes the unknown terrain height and the unknown building height. We assume that the height of the buildings does not vary a lot and the large height variations are mainly due to variations of the terrain height. We also assume that the terrain height is relatively constant in the close neighborhood of a building. Then, the main idea is to divide the interval of height variations into smaller intervals. On these smaller intervals the former method can be applied to determine the maximal region and the minimal region inside of which the building could lie. For each of these small intervals a function that looks for evidence in the image that the building lies in that region is defined. In order to compute this function lines are extracted from the image and the image lines which lie between the maximal and the minimal region are selected. Afterwards the length of the selected lines are summed up to compute the score for the given interval of height values. The score is integrated over both images. A high value indicates a high likelihood for the correct height of the building. This score is computed for each height interval.

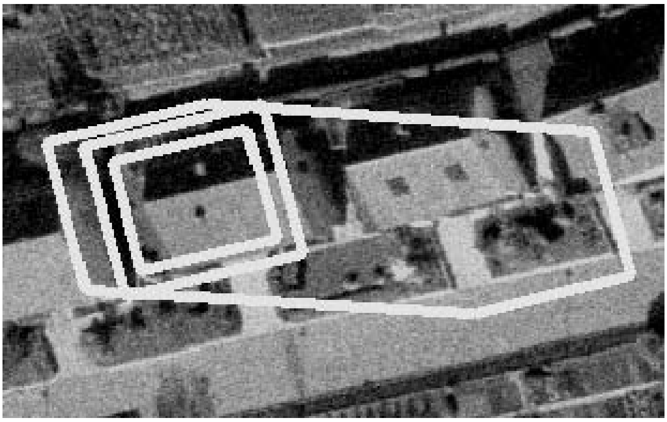


Figure 2.13: The pattern used in the localization process

Actually this process can be seen as moving the pattern (Figure 2.13) defined by the maximal and minimal region along a line in the image and counting for evidence for each position. In this way we get the graphic of the likelihood of having the building correctly localized for different height values (Figure 2.14 corresponding to the building from Figure 2.12).

Generally, a threshold of 70% of the largest peak can be used to select the peaks which might correspond to the correct building height. With this thresholding a large interval of height values is reduced to some individual values.

Unfortunately, not all peaks in the graph correspond to correct building locations. For instance, an edge corresponding to a road might introduce a false peak in the graph. Therefore, each peak has to be verified. This verification can be done by actually trying to find the building model which best describes the image data.

Since the building model generation process was designed to be able to deal with small height variations, there is no need to check every individual height value. If these height values can be grouped in small intervals then it suffices to check a value from each group. Therefore a k-nearest neighbor algorithm is applied for creating clusters of height values. The value of k is determined according to the variations that can be handled by the reconstruction procedure (size of the buffer used in the fitting algorithm, see Section 3.2.2).

For the building in Figure 2.12, by thresholding we get 8 height values. These 8 values can be further reduced to 5 clusters of height values by applying k-nearest neighbor classification. Figure 2.15 shows the two most likely locations of the building corresponding to $h = 224$ and $h = 253$.

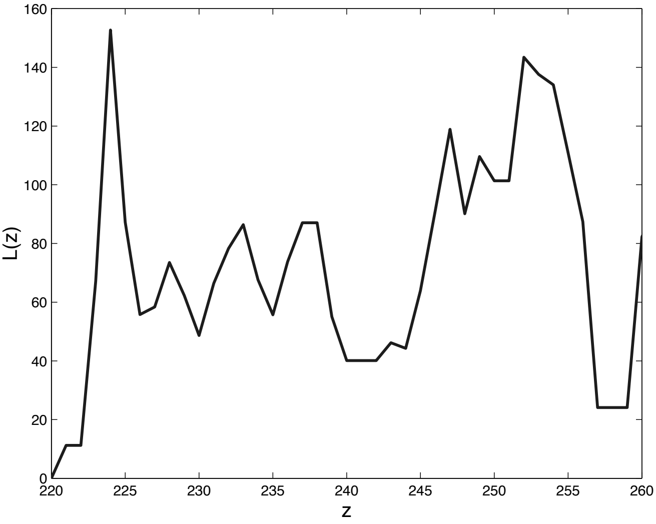


Figure 2.14: Likelihood of building location

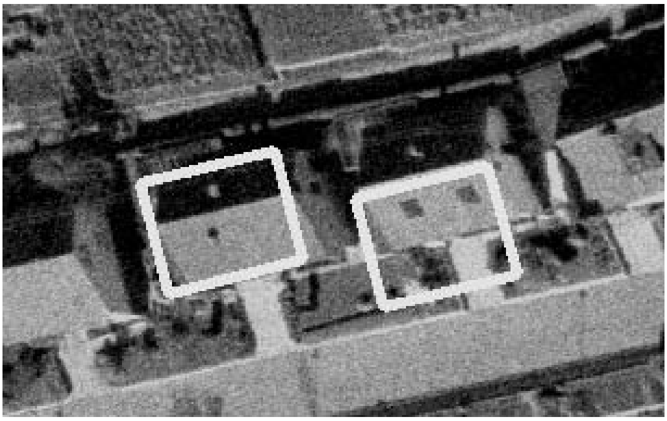


Figure 2.15: Possible locations of the building

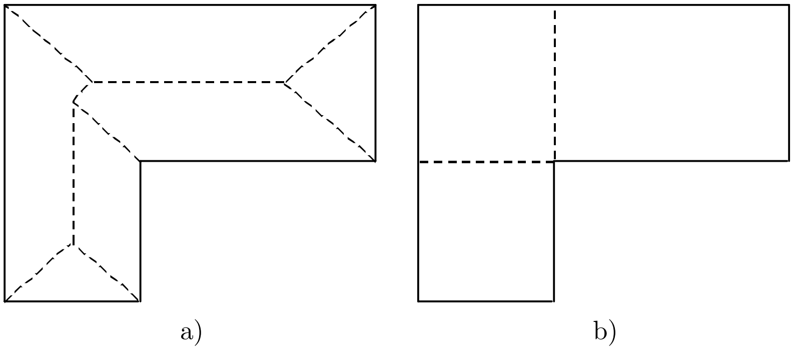


Figure 2.16: Ground plan segmentation a) Straight skeleton segmentation b) Rectangle segmentation

The height corresponding to the highest value of each cluster is used to generate some building hypotheses which are going to be verified by fitting them to image data. Hence the localization of the building in the image will only be completed when the correct building model is found.

2.3 Partitioning of a Building

Most complex buildings can be described as an aggregation of simple building types with a rectangular base. Hence, the actual building reconstruction can start by partitioning the building into simple building parts, which might correspond to the building primitives defined in the building library. The ground plan of the buildings provides useful information for this process. It often gives an indication of how to place these primitive models and can be used to derive hypotheses on the partitioning of the building.

Brenner [Brenner, 2000] also starts the building reconstruction by segmenting the ground plans. The straight skeleton of the ground plan is determined by assuming that all eave lines have the same height and all roof planes departing from these eave lines have the same slope. The roof planes are inclined towards the interior of the ground plan. The intersection of the planes yield ridge lines where two planes meet and points where three planes meet. These are projected back to the ground and the straight skeleton is obtained. The straight skeleton is unique, but unfortunately is not always the correct one. However, other possible segmentations of the ground plan can be derived from the skeleton. The number of segmentations are reduced by a discrete relaxation taking into account the labels of the edges (Figure 2.16a).

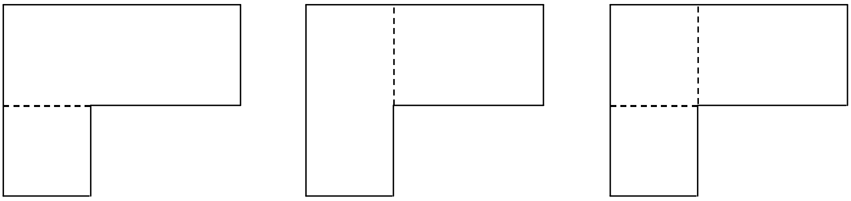


Figure 2.17: Multiple partitioning schemes

Our segmentation method is based on dividing the ground plans into rectangles. By extending the ground plan lines that intersect in concave corners a building ground plan can be segmented into small rectangles as shown in Figure 2.16b.

The partitioning can be described as a split-merge algorithm. In the first step, if the ground plan is not a rectangle, then it is segmented into rectangles, called partitions. A partition might correspond to a building primitive. But, it is possible that a group of partitions corresponds to a building primitive. Therefore the two or more partitions are merged together to form a bigger partition. Two partitions can be merged if they share a common edge.

A partitioning scheme can be defined as a subdivision of a building into disjoint partitions. Partitioning schemes are generated by sequential merging of the partitions. A building can have multiple partitioning schemes, more or less likely to occur in reality (Figure 2.17)

Figure 2.18 shows several building hypotheses generated on the basis of the segmented ground plan and the usage of building primitives with flat and gable roof.

2.3.1 Partitioning Rules

The problem with this approach is that a large number of partitions and partitioning schemes have to be treated even for a relatively simple ground plan. The number of partitioning schemes will virtually explode with increasing ground plan complexity. The number of partitions and consequently the number of partitioning schemes can be reduced by introducing some simple partitioning rules.

These rules can be acquired by learning from image interpretation. Within a study area, the true partitioning schemes of the buildings based on the ground plans were established and compared to the partitioning schemes resulting from the interpretation of images. This study led to the frequently occurring partitioning cases, which can be encoded in partitioning rules.

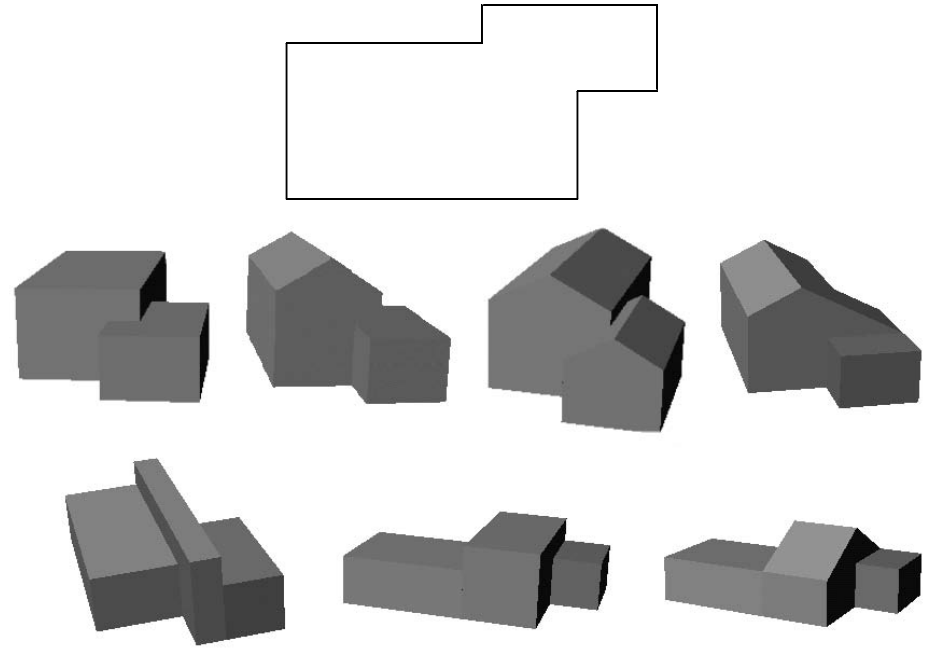


Figure 2.18: Different possible building models corresponding to a ground plan

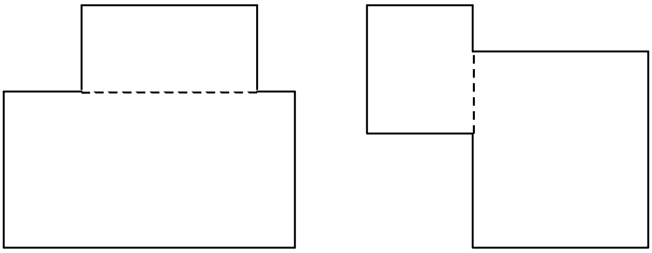


Figure 2.19: First partitioning rule

The simplest partitioning rule divides the building into two parts by extending two existing collinear edges of the ground plan (Figure 2.19).

Another partitioning rule splits a building part off by extending two existing rectangular lines, whereas the extensions are intersecting (Figure 2.20). In this case there are two partitioning possibilities. However, in most of the cases the larger rectangle has to be split off.

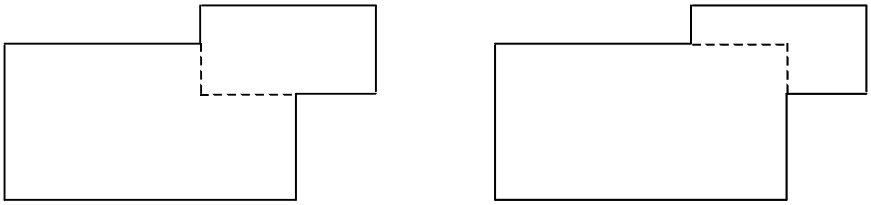


Figure 2.20: Second partitioning rule

Although, the partitioning rules are rules with high probabilities, in some cases they might not hold. Therefore, the algorithm should be able to backtrack in case of failure.

Chapter 3

Generation of Building Hypotheses

This chapter describes the generation of building hypotheses by instantiation of the building primitives from the building library. This instantiation requires some initial values for the parameters of the building primitives. Therefore, some 3D information is derived. The 3D corner extraction by stereo matching is presented.

The generated building hypotheses are refined by fitting to images. There are a number of existing techniques that fit object models to image data. Some of them are briefly described in the second part of the chapter. Finally the fitting method used in this thesis is presented. This method is used both for fitting building primitives in the hypotheses generation process and for fitting complete building models in the final hypothesis verification step.

3.1 3D Feature Computation

The building hypotheses generation process requires extraction of initial values for the parameters of the primitives. For initializing the height values some 3D information from images are extracted. This 3D information consists of 3D corners and 3D lines obtained by matching 2D features extracted from the stereo images.

3.1.1 3D Corner Generation

The corners were selected as basic primitives for the reconstruction process for several reasons.

- Simplicity: The corners encode particular geometric relationships, thus matching them involves less ambiguity than matching individual image features, since there are fewer possible alternatives and more information to judge a match. In addition, there are fewer corners than image features.
- Robustness: Corners are less likely to be completely occluded as opposed to single features. Due to the building geometry, the corners are relatively stable against partial occlusions. Even if a corner point is occluded, its position can be found by intersecting the corresponding line segments. Also, the position of a corner is generally measured with more precision than the position of an isolated feature. Hence, the 3D reconstruction of a corner is more precise.

First, image features, namely interest points and line segments, are extracted from the images, then 2D corners are generated and these 2D corners are later matched leading to 3D corners.

2D Corner Generation

Many feature extractors have been proposed [Deriche and Giraudon, 1993] [Burns et al., 1986] [Smith and Brady, 1997] [Steger, 2000]. In our system the image features are extracted using the Förstner operator [Foerstner, 1994], which can be used for interest points as well as for line segments extraction. This operator is gradient based and it can determine point and line locations with sub-pixel accuracy.

Consider the average squared gradient described by the following matrix:

$$N = \nabla g \nabla g^T = \begin{bmatrix} \sum g_x^2 & \sum g_x g_y \\ \sum g_y g_x & \sum g_y^2 \end{bmatrix} \tag{3.1}$$

where $\nabla g = (g_x, g_y)$ is the gradient. If the two eigenvalues of the matrix N are large, then there is an important change of gray level. The trace of this matrix can to be used to select image pixels which correspond to image features.

$$tr(N) = \lambda_1 + \lambda_2 \tag{3.2}$$

where λ_1, λ_2 are the eigenvalues of the matrix N. If $tr(N)$ is higher than a threshold then the pixel is considered as a feature pixel.

Next, the interest points have to be distinguished from pixels belonging to edges. This can be done by ratio $v = \frac{\lambda_2}{\lambda_1}$ which yields the degree of orientation or an isotropy. $\lambda_2 = 0$ indicates a straight edge, while $v = 1$ thus $\lambda_2 = \lambda_1$ indicates a gradient isotropy caused by a corner or a blob. The corner response function is given by:

$$q = \frac{4 * det(N)}{tr^2(N)} = \frac{4\lambda_1\lambda_2}{(\lambda_1 + \lambda_2)^2} \tag{3.3}$$

This value has to be larger than a threshold for a pixel representing a corner. Sub-pixel precision is achieved through a quadratic approximation.

To avoid corners due to noise, the images can be smoothed with a Gaussian kernel. But instead of doing this smoothing on the original images, it is done on the images of the squared image derivatives contained in matrix N.

Figure 3.1 shows the extracted interest points and line segments for some buildings. The features were extracted only from the delineated area of the buildings. In practice, often far too many features, especially interest points are extracted. Therefore, it is often interesting to first restrict the numbers of corners before trying to match them. One possibility consists of selecting the interest points with a value above a certain threshold. This threshold can be tuned to yield the desired number of features. However, there are interest points which are caused by the roof or ground texture and hence are not relevant for the 3D reconstruction task. Another possibility is to work with more complex features, obtained by grouping from the simple features. These features can be the corners.

A corner is defined as a corner point and a set of corner lines, whose intersection point is the corner point. A 2D corner is generated based on the proximity relation between an interest point and a set of lines or between a set of lines if the interest point is not detected.

In determining the proximity relation between a point and a line, three distances are computed [Beveridge, 1993]: 2 end-point distances and 1 projection distance (Figure 3.2a). For the proximity relation between two lines 8 distances are computed: 4 end point distances and 4 projection distances (Figure 3.2b).

If the projection of the corner point lies outside of the segment then the projection distance is ignored. If the smallest of the computed distances is below a predefined threshold, then the point and the line, and the two lines, proximate. Finally, all the lines are extended to the corner points found at their ends. In this way for every point a set of corresponding lines is determined. If there are parallel lines in this set of lines then, for each line from the set of parallel lines, a 2D corner hypothesis is generated including the corner point, this line, and the rest of the lines, which are not parallel to this line.

The generated 2D corners are classified as wall corners or roof corners (see chapter

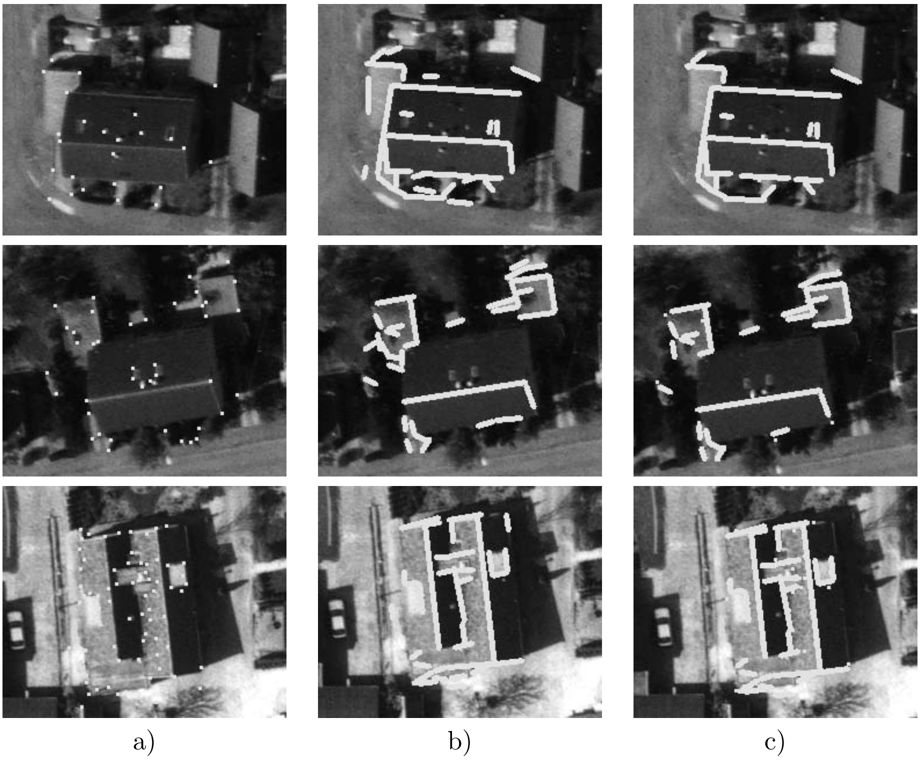


Figure 3.1: 2D Corner generation. a) Extracted interest points b) Extracted line segments c) Generated 2D corners

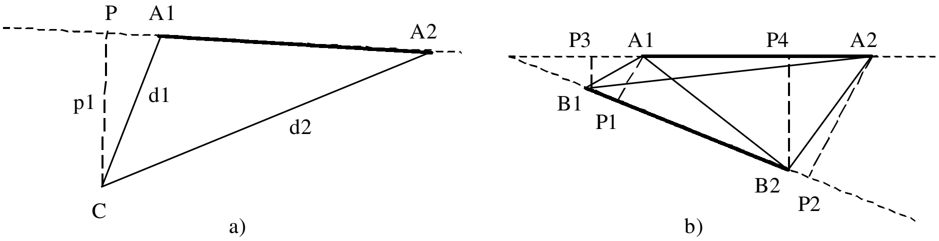


Figure 3.2: a) Proximity relation between a point and a line b) Proximity relation between two lines

2.2). The wall corners are assigned the label of the map corner which they might correspond to.

2D Corner Matching

The 2D corners from different images have to be matched in order to get 3D corners. To cope with the combinatorial complexity of the matching, many constraints have been incorporated in the matching process. To assure invariance to lightning conditions, the intensity of features are not considered in the matching. One of the reasons why the corners were selected as a basis for the reconstruction process is because the corner points strongly restrict the search for correspondences using epipolar geometry. The constraints used for the corner point matching are:

- Corner label: a corner labelled as a wall corner in one of the images can only be matched with a corner corresponding to the same wall corner, or with a roof corner of the other image. However, roof corners can be matched with any kind of corner.
- Epipolar geometry: The epipolar constraint is applied to restrict the search for correspondences along the epipolar line.
- Height: The 3D points obtained by triangulation from the two 2D corner points must have a height between some extreme values. This problem is identical to the determination of the disparity search range along the epipolar line.
- Ground plan of a building: The 3D points must lie inside or sufficiently close to the ground plan of the building defined in the map.

The order in which the constraints are applied was chosen in such a way as to reduce the total amount of computation required for matching two corner points. The first constraint is the simplest one, the corner label. The second constraint is the epipolar constraints. This epipolar constraint can be represented algebraically by the essential matrix when the intrinsic parameters of the cameras are known, and by the fundamental matrix otherwise. Therefore, it requires the computation of the epipolar geometry between the two images (see appendix A.2). In the ideal case the epipolar equation is given by:

$$p_{xr}^T F p_{xl} = 0 \tag{3.4}$$

where p_{xl} and p_{xr} are two corresponding image points, and F is the fundamental matrix. The epipolar lines corresponding for some selected points are presented in Figure 3.3.

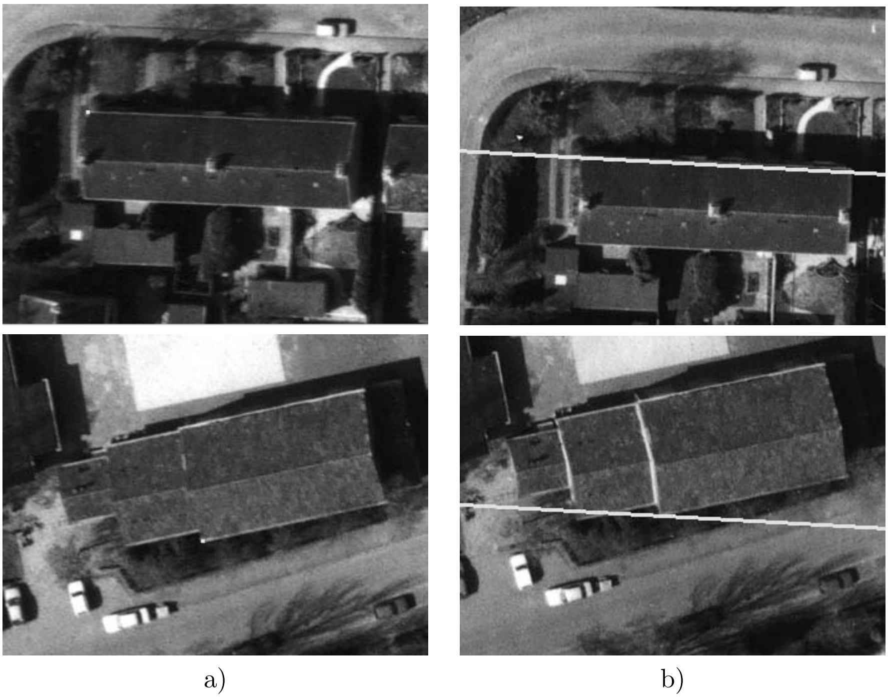


Figure 3.3: Epipolar constraint a) Selected point b) Epipolar line

Due to uncertainty, the previous equation does not have exactly zero on its right-hand side. The epipolar equation is modified to be:

$$p_{xr}^T F p_{xl} < \epsilon \tag{3.5}$$

where ϵ depends on the accuracy of the orientation parameters. This modification can be interpreted as searching in the vicinity of the epipolar lines and not just at their theoretical locations.

If the first two constraints are satisfied then the 3D point is computed by triangulation and the other two constraints are checked for. If there are still multiple possible matches, then the local neighborhoods of the corners are compared through intensity cross-correlation. As a neighborhood, a small window of $(2N+1)\times(2N+1)$ pixels centered around the corner can be taken. For two points (x, y) and (x', y') in the left and right images, respectively, the similarity measure is given by:

$$C = \frac{1}{\sigma\sigma'} \sum_{i=-N}^{N} \sum_{j=-N}^{N} (I(x-i, y-j) - \overline{I})(I'(x'-i, y'-j) - \overline{I'}) \tag{3.6}$$

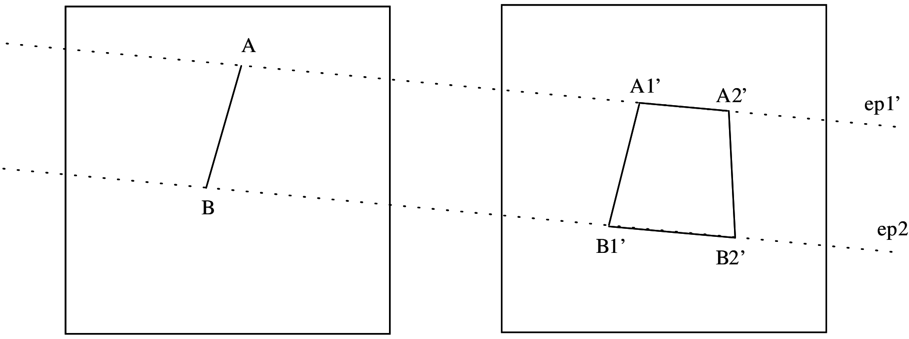


Figure 3.4: Epipolar constraint for matching line segments

where I and I' are the intensity values at a certain point, $\overline{I}$ and $\overline{I'}$ are the mean intensity values of the considered neighborhood and σ and σ' are the standard deviations of the intensity values. C varies from -1 for completely uncorrelated windows to 1 for identical windows. Typically $N = 3$ which yields 7 x 7 windows. Finally, points with high similarity score are chosen as correspondent points.

3.1.2 3D Lines Generation

The problem of matching line segments is slightly more complex than that of matching points, because the line segment extraction algorithm often produces different results in the two images. There will be gaps and fragmentation, missing line segments, erroneous additional segments. Hence, two line segments generally do not correspond globally and only contain a subset of homologous points.

When matching lines over two views, there is a weak overlap constraint for line segments due to epipolar geometry. The line segment in one view must lie at least partially within a beam defined by the epipolar geometry and the height constraints of the building [Zhang, 1994]. Consider Figure 3.4. By the epipolar constraints for points, the match for point A must lie on the epipolar line $ep1'$, defined by A. Similarly, a match for point B must lie on $ep2'$. By knowing the height range of a building, the search space can be reduced to the segments on the epipolar lines defined by the extreme height values. Hence, the search space for matching line segments is limited to a four-corner polygon. Each line segment that is at least partially inside of this polygon can be matched with the initial line segment.

A simplified epipolar constraint for matching pairs of line segments imposes that the epipolar line passing through the midpoint of one segment intersects the other

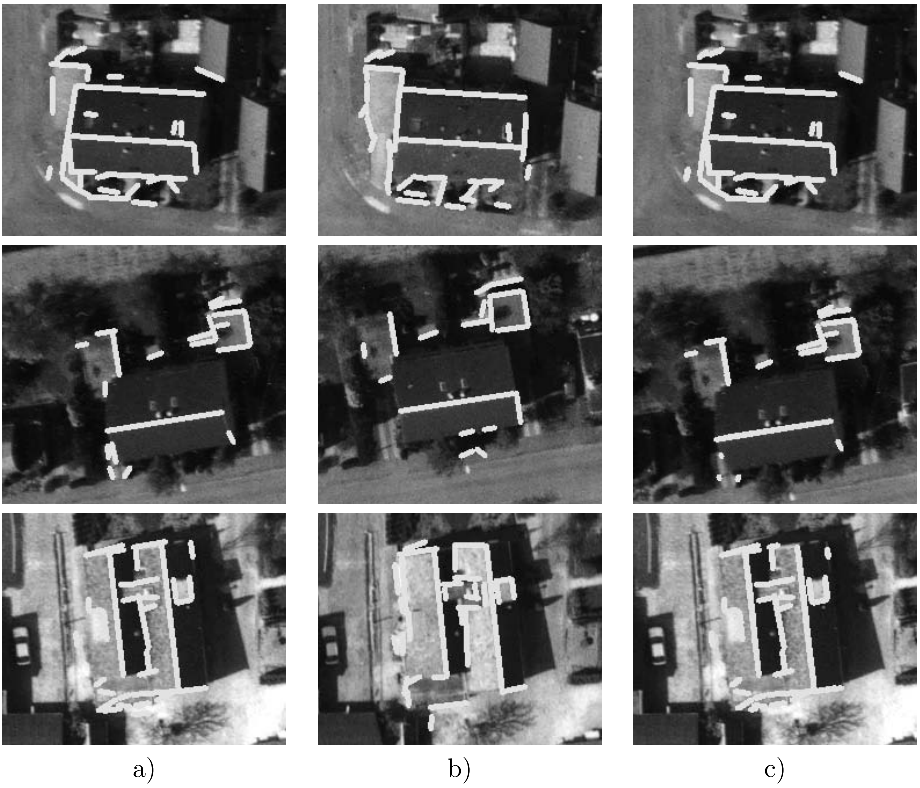


Figure 3.5: Matching line segments. a) and b) Extracted line segments in the left and right image respectively c) Reconstructed 3D line segments projected back into the left image

segment [Ayache and Faverjon, 1987].

Figure 3.5 shows the results of a putative line segment matching for two scenes. The only constraint used for the matching was the epipolar constraint. In the first scene there are 30 and 33 line segments detected in the left and right images, respectively, and 121 line pairs are matched. For the second scene there are 33 and 21 line segments detected in the left and right images, respectively, and 93 line pairs are matched. So, the epipolar constraint reduces the search complexity to more than 1/5. But still there are a lot of mismatches.

Another problem of the line segment matching is due to the lines that are parallel to the epipolar lines forming the epipolar beam. These lines cannot be reconstructed in 3D. This case is shown in Figure 3.3 where the epipolar line is parallel with one

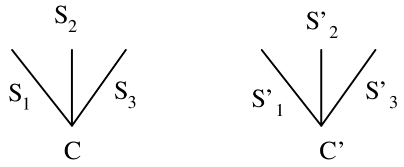


Figure 3.6: Order constraint

of the edges of the building.

The complexity of the line matching is somewhat simplified in case of lines grouped into corners. The constraints used for matching line segments belonging to corners are:

- Epipolar geometry: Even if only a small number of line segments correspond to a corner, the epipolar geometry still can be used to reduce the possible correspondences between corner lines.

  In this way, all possible correspondences between the line segments of the 2D corners are determined and the corresponding 3D line segments are computed. From this set of 3D lines corresponding to a 3D corner, 3D corner hypotheses are generated, which have to satisfy two additional constraints.

- Uniqueness: This constraint imposes that one line segment from the first image has, at most, one corresponding line segment in the second image (a symmetric constraint applies to a line segment of the second image).

- Order: This constraint assumes the preservation of the order of corresponding lines from the two images (Figure 3.6). If S_1 matches S'_2 then S_2 cannot match S'_1.

Results of the corner matching are shown in Figure 3.7. For the first building, 17 corners are generated in the first image and 39 in the second image. By matching them we get 18 3D corners with 31 line segments. This compared to the 121 line segments obtained by matching the 2D line segments using only the epipolar constraint represents a good reduction. For the second building, there are 29 corners in the first image and 28 corners in the second image and finally we get 18 corners with 35 line segments. There are still many spurious matches and many 3D building corners are not reconstructed at all. Therefore, it would be really difficult to start the building reconstruction process from the generated 3D corners. However, these 3D corners constitute important data for higher level processes.

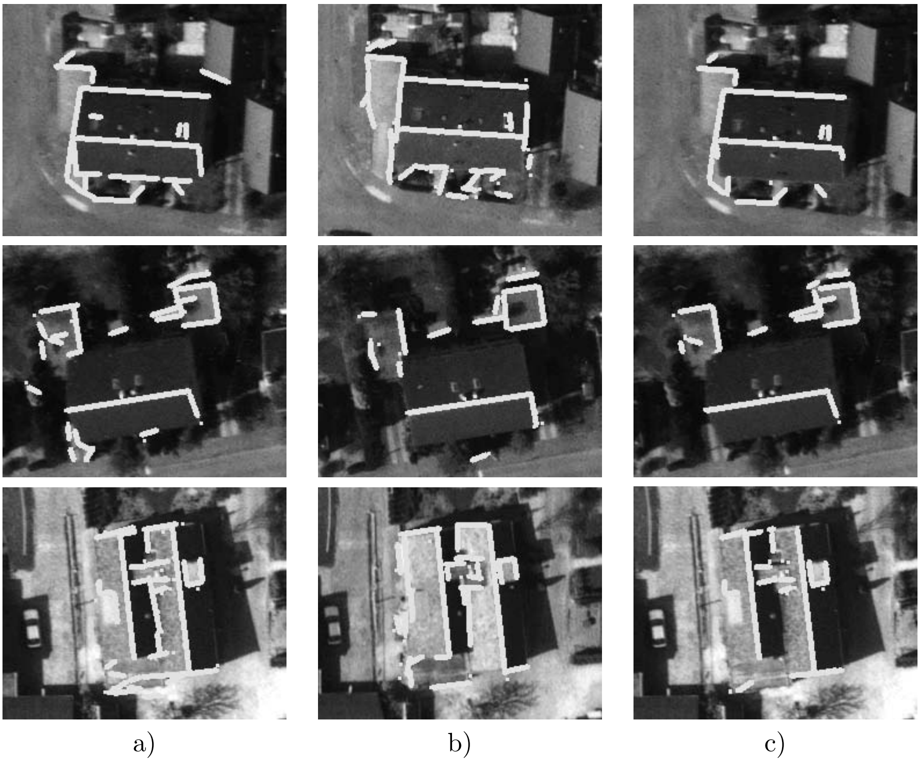


Figure 3.7: Matching corners. a) and b) 2D corners generated in the left and right image respectively c) Reconstructed 3D corners projected back into the left image

3.2 Fitting Building Models

3.2.1 Model Fitting Strategies

The aim is to locate a 3D object whose geometrical properties are fixed, but whose appearance can vary and this variance is encoded in a few parameters. To find the best fit of an instance of such a model to an image, we must find the model parameters, which best match the model to the image. The procedure for estimating the unknown parameters is known as model fitting.

Many approaches for model fitting have been proposed in literature. Different optimization methods [Press et al., 1992] have been employed to optimize the numerical estimate of the model parameters. Some of them are based on reducing the problem to a least-squares one. Projection from 3D to 2D is a nonlinear problem. Therefore, the solution can be based on Newton's method of linearization and iteration to perform a least-squares minimization.

The ***Gauss-Newton method*** gives a very efficient means of converging to a solution, if a sufficiently good initial guess is provided. Rather than solving directly for the vector of parameters p, Newton's method computes a vector of corrections x to be subtracted from the current estimate of p on each iteration. If $p^{(i)}$ is the parameter vector for iteration i, then:

$$p^{(i+1)} = p^{(i)} - x \tag{3.7}$$

Given a vector of error measurements e between components of the model and the image, the goal is to solve for an x that minimizes this error. Based on the assumption of local linearity, the effect of each parameter correction x_i on an error measurement will be x_i multiplied by the partial derivative of the error with respect to that parameter. Therefore, the following equation has to be solved for x:

$$Jx = e \tag{3.8}$$

where: J is the Jacobian of the objective error function e and

$$J_{ij} = \frac{\partial e_i}{\partial x_j} \tag{3.9}$$

Each row of this system states that each measured error e_i should be equal to the sum of all changes in that error resulting from the parameter corrections x. If all these constraints can be simultaneously satisfied and the problem is locally linear, then the error will be reduced to zero after subtracting the corrections.

The overdetermined system is solved by minimizing the term $\|Jx - e\|^2$ in a least-squares sense. This minimization has the same solution as the normal equations:

$$J^T Jx = J^T e \tag{3.10}$$

Lowe [Lowe, 1991] proposed to measure the pose error e perpendicular to image line features. Successful application of Newton's method requires starting with an appropriate initial choice for the parameters and, even in this case, there is still a risk of convergence to a false minimum.

Lowe showed that the ***Levenberg-Marquardt method*** is well suited to stabilize the solution. An additional parameter λ is introduced with a weighting matrix W. This matrix accounts for the standard deviations of the parameters. This leads to the minimization of

$$\|Jx - e\|^2 + \lambda^2 \|W(x - d)\|^2 \tag{3.11}$$

where d represents a vector of default parameters. The strategy of varying the parameter λ according to the residual error allows control over the behavior of the method between pure gradient descent (large value of λ) and the Newton-Raphson method (small value of λ).

Sullivan developed a model fitting procedure based on ***Powell's direction set method*** for determining the pose of cars [Sullivan, 1992]. Constraining the position of a car to the ground plane, the pose is given by only two dimensions of translation and one of rotation about the vertical axis. The alignment between the model and the image data is tested by iconic evaluation. The evaluation function defines a scalar function in three dimensions. In general, peaks in these three evaluation functions indicate likely matches between the model and the image.

An exhaustive search of the evaluator surface over three dimensions is computationally too expensive. Therefore Powell's direction set method is used. This method successively decomposes the problem into three separate one-dimensional searches. For each sample the model is displaced by an appropriate amount (in the object coordinate frame), instantiated into the image, and its fit to the image is evaluated. The best score and its position are noted and the process is repeated for the other two variables, each time starting from the same initial pose. When all three dimensions have been searched the pose having the highest score found is adopted. It then becomes the initial position for the next iteration, and the search coordinate frame is changed accordingly. If no higher score is found then the three ranges of the search are reduced, to be equal to the previous sampling interval. The process is repeated until all the sampling intervals fall below criterion values.

Snakes or ***active contour models*** have been introduced by Kass, Witkin, Terzopoulos [Kass et al., 1987]. Fitting a snake to data was defined as an optimization problem that sought a minimum energy boundary subject to some constraints. A useful formulation is to consider that the total energy is a sum of three components:

1. internal contour energy characterized by stretching and bending the contour itself.
2. image energy that characterizes how the contour fits to the image intensity or gradient
3. external energy due to constraint forces.

A snake is a curve with an intrinsic parameter s and a dynamical evolution indexed by time t. Its equation can be expressed in 2D as:

$$\vec{v}(s,t) = (x(s,t), y(s,t))_{0 \leq s \leq |C|} \tag{3.12}$$

and in 3D as:

$$\vec{v}(s,t) = (x(s,t), y(s,t), z(s,t))_{0 \leq s \leq |C|} \tag{3.13}$$

where $|C|$ represents the length of the curve.

In order to have this curve evolve in time, external forces derived from a field of energy $\vec{F} = -\nabla E$ are used. The problem is to find the function that minimizes the energy defined as follows:

$$E_{contour} = \int_C (E_{internal} + E_{image} + E_{constraints})\, ds \tag{3.14}$$

Fua [Fua and Leclerc, 1990] used generalized snakes for fitting polyhedral object models to images. A polygonal snake is modelled as a list of vertices.

If the snake is to converge towards the edges in the image, then the image energy can be considered as the energy issued from the image gradient:

$$E_{image}(C) = -\frac{1}{|C|} \int_C | \bigtriangledown I(\vec{v}(s,t))| ds \tag{3.15}$$

where I is the image grey level.

E_{image} is computed by projecting the curve into the images, computing the image energy for each projection and summing up these energies. In practice the image energy is computed by integrating the gradient values $|\nabla I|$ in precomputed gradient images along the line segments that connect the polygonal vertices.

The geometrical relationship between the connected vertices of the polygonal model are described by a regularization term $E_{reg}(C)$. Then the total energy can be defined as:

$$E_{total}(C) = E_{image}(C) + E_{reg}(C) \tag{3.16}$$

To perform the optimization, the Euler-Lagrange equation of dynamics is derived:

$$\frac{\partial E_{total}}{\partial S} + \alpha \frac{dS}{dt} = 0 \tag{3.17}$$

where α is the viscosity and controls the allowed corrections of the list of vertices $S = (x, y, z)$ of the polygonal model.

The following formula is used to compute the viscosity:

$$\alpha = \frac{\sqrt{2n}}{\triangle_p} \left| \frac{\partial E_{total}}{\partial S} \right| \tag{3.18}$$

where n is the number of vertices and $\triangle_p$ is the initial step size. The viscosity is computed at the start of the optimization and progressively increased as needed to ensure a monotonic decrease of the energy of the snake and ultimate convergence of the algorithm.

At each iteration the system has to solve the following differential equations derived from equation (3.17):

$$(K + \alpha I)V - t = \alpha V_{t-1} - \left. \frac{\partial E_{image}}{\partial V} \right|_{V_{t-1}} \tag{3.19}$$

where K is a sparse matrix, and V stands for either X or Y, the vectors of the x and y vertex coordinates.

Furthermore hard constraints can be integrated in the optimization procedure for enforcing the geometric relationship between vertices. Examples of constraints that can be applied for a polygonal object model are: vertices have the same height, edges of the polygon are parallel or perpendicular.

These generalized snakes were used for automatic extraction of road and building edges from cluttered aerial images [Fua, 1997]. In case of building models, the curves don't have to be forced to be smooth. Therefore the regularization energy can be neglected. Thus, each iteration of the optimization amounts to solving the linear equation:

$$\alpha(S_t - S_{t-1}) = \left. \frac{\partial E_{image}}{\partial S} \right|_{S_{t-1}} \tag{3.20}$$

Analysis of the Fitting Strategies

In this paragraph the characteristics of the above presented fitting algorithms are discussed.

Some methods for fitting 3D models to images are based on the extraction of image features, matching these features to geometrical components of the model (for instance closest projected model line) and minimizing the distance between image features and model features. The main problem of these methods is that they are highly dependent on accurate matches between image features and model features. They offer fast convergence at the price of smaller pull-in range. Lowe's algorithm belongs to this category of fitting methods. It requires edge detection. Only edge pixels stronger than a preset threshold are used in the parameter estimation process.

Sullivan uses directional derivatives extracted from images to refine the parameters without edge extraction. But the search and voting process employed, produces oscillations of the parameter values.

Snakes are the most intuitive fitting methods. By defining a proper energy function for the snake, a direct relationship between the estimation of the parameter values and pixel gradient can be achieved. The object models used are described by their vertices (B-rep representation), which allows quite general models. Constraint describing geometric relationships between vertices of the model can be integrated in the fitting process. The drawback is that the optimization algorithm for estimating the parameters is computationally expensive, since the derived image energy gradients only indicates the direction in which the vector of parameters has to change. It takes a large number of iterations to accurately determine the amount of change required to obtain the best fit. To improve the speed, the value of $\triangle_p$ can be increased, but this will reduce the success rate of the fitting.

3.2.2 Fitting of a Building Primitive

The method used in this work is a modification of Lowe's fitting algorithm described above. It was developed by Vosselman and used for reconstructing 3D buildings [Vosselman and Veldhuis, 1999] and industrial piping installation [Tangelder et al., 1999]. The method overcomes the above mentioned drawback of Lowe's algorithm. The edge extraction step required by Lowe's algorithm is eliminated. Pixels regardless of their gradient value will contribute to the fitting. Hence, a direct relationship between the parameter estimation and the gradient can be achieved.

The approach for fitting 3D building models to an image is based on projecting the model into the image and finding the parameters of the model that maximizes some measure of the goodness-of-fit between model projection and image. In most cases it is possible to solve for all unknown parameters of a building model from fitting to a single image. However, the accuracy of the parameter estimation can be substantially improved by simultaneously fitting the model to images taken from different viewpoints. The method presented here can be used in either situation.

Initial Approximation

The application of this fitting algorithm requires initial approximate values for the model parameters. These approximate values can be obtained from the map and 3D information extracted from images.

In the initial approximation the x, y coordinates and the orientation of a building primitive are given by the ground plan of the building. The width and length parameters are the width and the length of the rectangle corresponding to the ground plan of the building partition.

The height of the building primitive is computed taking into account the heights of the reconstructed 3D corners of the building partition. For this purpose a simplified ranking scheme was developed. The height is given by the average of the 10% minimal and maximal values as estimations of the minimum and maximum.

For computing the height of the ridge of a symmetrical gable roof primitive more cases can be distinguished:

1. If the top lines are detected in both images and the 3D line can be reconstructed, then the height of the ridge is the height of the reconstructed 3D top line.
2. If the top line is detected in one image and in the other image an approximate position of the projected ridge can be deduced, then the 3D ridge can be reconstructed by matching these two line segments. An approximate position of a projected ridge in an image can be deduced by taking into account the symmetry of a gable roof, if the two base lines of the gable roof are detected in the image.
3. Otherwise a default value is considered for the height of the ridge.

For a non-symmetrical gable roof initial values for the height of the ridge and the distance from the roof reference point to the ridge base point are also required. The height of the ridge can be computed only if the top lines are detected in both images. Otherwise a default value has to be considered. For computing the distance from the roof reference point to the ridge base point dr, three cases can be distinguished:

1. If the top lines are detected in both images and the 3D line can be reconstructed, then dr can be computed.
2. If a top line is detected in one image then dr is estimated by projecting the reference point of the roof into this image and computing the distance from this point to the top line.

3. If no top line is detected, then a default value is considered for dr.

These initial values are influenced by uncertainties of the knowledge sources. The uncertainties are due to the accuracy of the GIS map, the roof extensions, and the estimated height values as detailed in chapter 2.2.

Precise Estimation

A more precise estimation of the parameters is obtained by fitting. This algorithm fits the edges of the projected wire-frame of the model to gradients of the pixels from both images simultaneously. The fitting method is described as an iterative least-squares algorithm. It estimates the changes of the parameter values that have to be applied in order to minimize the square sum of the perpendicular distances of the image pixels to the nearest wire-frame edge.

An observation equation is set up for each image pixel within some range of a wire-frame edge. The linearized observation equation for a pixel j can be written as:

$$E\{\Delta u_j\} = \sum_{i=1}^{K} \frac{\partial u_j}{\partial p_i} \Delta p_i \tag{3.21}$$

where Δu_j is the observed perpendicular distance of the pixel to the nearest edge of the wire-frame, p_i are the model parameters, K is the number of parameters, and Δp_i are the changes of the parameters that have to be estimated.

Lowe introduces an observation equation for each edge pixel within a buffer of a projected wire-frame. This assumes edge extraction that requires selection of a threshold. Also, weak edge pixels are not used in the fitting if their edges are below the threshold. To avoid these problems, in Vosselman's approach each pixel is considered. However, in order to ensure that the pixels with higher gradients dominate the parameter estimation, the squared gradient of the pixel can be used as a weight to its observation equation.

$$W\{\Delta u_j\} = \left(\frac{\partial g_j}{\partial u_j}\right)^2 \tag{3.22}$$

The gradient used in the weight function is calculated in the direction u_j perpendicular to the edge of the wire-frame. The partial derivatives of the distances with respect to changes in parameters are estimated numerically by finite differences.

$$\frac{\partial f}{\partial x} = \lim_{\delta x \to 0} \frac{f(x + \delta x) - f(x)}{\delta x} \tag{3.23}$$

The steps of the fitting algorithm can be summarized as follows. The method starts with the projection of the wire-frame model into the images using the know interior and exterior orientation parameters and the initial shape and pose parameters of the model. Next the partial derivatives of the parameters are computed. For these computations, for each parameter a new object model is generated with the value of the parameter slightly changed. All these new models are projected into the images. Afterwards for each pixel within the buffer around a model edge, the partial derivatives are computed and the observation equation (3.21) is added to the normal equation system. The derivative ($\partial u/\partial p_i$) of the distance with respect to specific parameter p_i is computed by using equation (3.23) between the distance of the pixel to the original model and the distance of the pixel to the nearest edge of the model with a slightly changed value for parameter p_i. Next the normal equation system is solved and the resulting changes are applied to the values of the parameters. This procedure is repeated until the residuals of the last two iterations are almost equal.

$$\frac{\sigma_k^2}{\sigma_{k-1}^2} \approx 1 \tag{3.24}$$

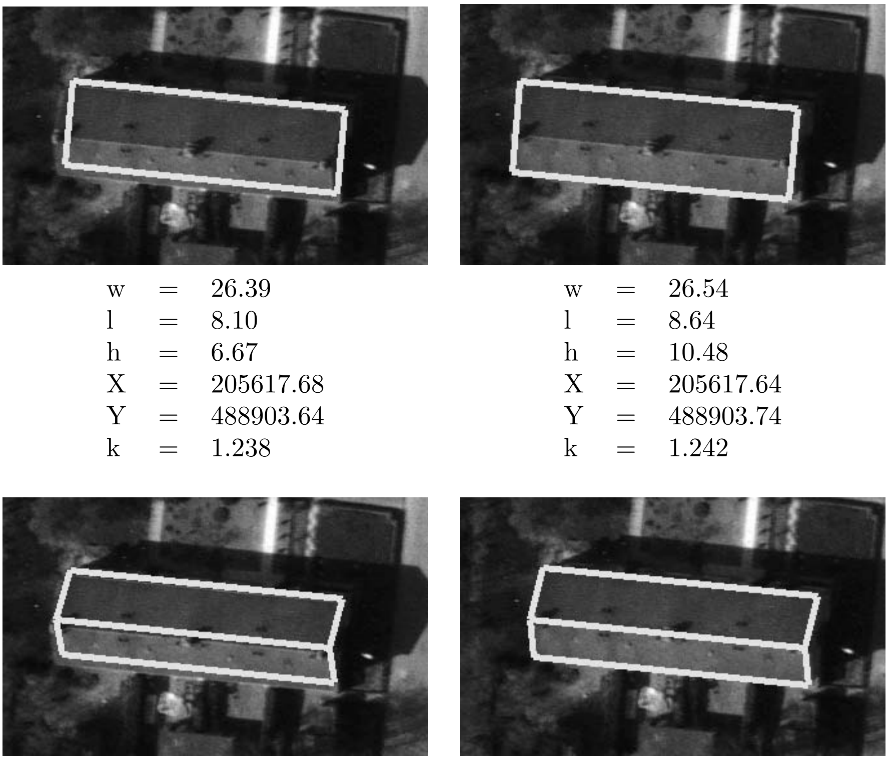
where σ_k is the residual after k iterations. The residual is given by:

$$\sigma^2 = \frac{\sum_i e_i^2}{n-p} \tag{3.25}$$

where $e = y - A\hat{x}$ and $y = Ax$ is the normal equation of the least-squares system.

The main advantage of this fitting algorithm is the fast convergence. The algorithm usually converges after a couple of iterations. The processing time depends a lot on the initial values of the parameters. Good initial values reduces the time. In [Vosselman and Veldhuis, 1999] some implementation aspects are presented that can speed up the algorithm. The idea is to start with a large buffer and then to reduce the size of the buffer after each iteration of the least-squares estimation. In the first few iterations with a large buffer size, the buffer is subsampled with profiles perpendicular to the edges. Observation equations are set up only for pixels on these profiles. The point density is increased each time the buffer size is reduced. In this way a large initial buffer size can be combined with a fast and precise fitting.

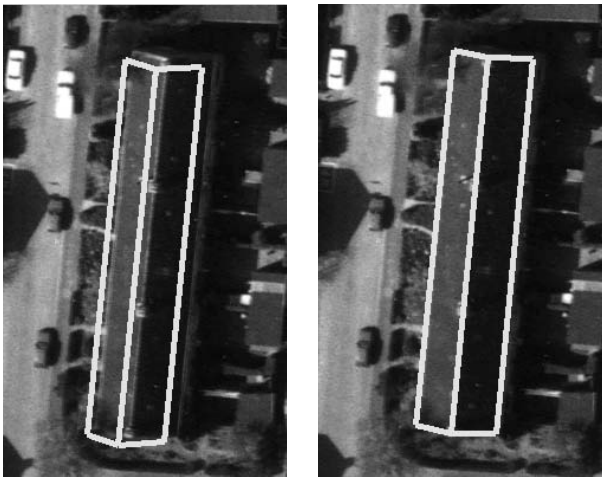
Figure 3.8 shows some examples of fitting building primitives. The flat roof building required 9 iterations, while the gable roof building needed 7 iterations. Note that a flat roof building was fit reasonably well where the real building is a gable roof building. Another example of fitting is shown in Figure 3.9. The X, Y coordinates and the orientation in XY plane (k) are quite stable, the width (w) and the length (l) change slightly, while the height (h) and height of the ridge (hr) vary a lot. Actually the value of the width and the length parameters increases as we expected. That is because they were initialized from the ground plan defined in the map which is smaller than the roof base. The results of the fitting are usually



w	=	26.39	w	=	26.54
l	=	8.10	l	=	8.64
h	=	6.67	h	=	10.48
X	=	205617.68	X	=	205617.64
Y	=	488903.64	Y	=	488903.74
k	=	1.238	k	=	1.242

w	=	26.39	w	=	26.51
l	=	8.10	l	=	8.54
h	=	6.67	h	=	9.75
X	=	205617.68	X	=	205617.71
Y	=	488903.64	Y	=	488903.73
k	=	1.238	k	=	1.237
hr	=	5.31	hr	=	2.56

Figure 3.8: Fitting of a building primitive Left: Initial values. Right: results of fitting.



w	=	8.09	w	=	8.44
l	=	39.32	l	=	39.43
h	=	12.75	h	=	9.21
X	=	205618.13	X	=	205618.02
Y	=	488903.64	Y	=	488949.17
k	=	1.237	k	=	1.238
hr	=	1.29	hr	=	2.15

Figure 3.9: Fitting of a building primitive. Initial building primitive and the fitted primitive

good. Most of the model edges are aligned correctly by the fitting algorithm. Misaligned edges can appear due to the presence of other objects in the neighborhood and in cases of fitting small building primitives.

The method does not require segmentation of the images, so no threshold has to be specified. It can be applied to multiple images. If multiple images are available then the performance of the fitting will improve.

3.2.3 Fitting of a CSG Tree

In the final verification step the CSG tree describing a building will be fit to the image data. In the previous stages of the building reconstruction process, the building models corresponding to different building partitions were treated as isolated objects. These models can be further refined if contextual information is utilized. The fact that the building models contained in the CSG tree form a complex building can be seen as contextual information. Between the building

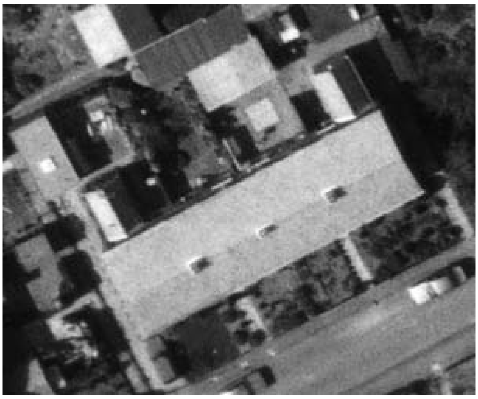


Figure 3.10: Complex building formed from a main part and a small shed

models of a complex building, many geometric relationships can be identified, which constitute very valuable information. Therefore, a global fitting algorithm can perform a simultaneous adjustment of the building models contained in the CSG tree taking into account the geometric relationships between them.

The geometric relationships between building models can be represented by constraints. In the parameter estimation process, these constraints mean that the parameters of different building models are correlated with each other. The usage of the constraints reduces the degree of freedom of some parameters; therefore, the precision of the parameter estimation is increased.

Consider the building shown in Figure 3.10, composed from a main part and a small attached shed. The estimation of the parameters of the small building extension can cause problems. By imposing the constraint that the shed is connected to the main building, its parameters can be estimated more precisely, since more information can be derived from the main building.

In our building reconstruction system the following types of constraints are used:

- Parameter constraints: establishes a relation between two parameters of two building models. For example, two building models have the same orientation.
- Connection constraints. One edge of a building model lies on one of the edges of the other building model (Figure 3.11a).
- Corner constraints. Two building models share a common corner (Figure 3.11b).

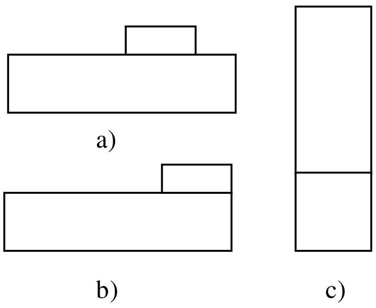


Figure 3.11: Constraints types. a) Connection constraints b) Corner constraints c) Extension constraints

- Extension constraints. Two building models share a common edge (Figure 3.11c).

Formally, the constrained optimization problem can be described as follows. Given a least-squares system with K parameters $\{p_1, p_2, ..., p_K\}$, we want to solve it under a set of m constraints $C = \{c_1, c_2, ..., c_m\} = 0$.

An efficient and accurate solution to this problem is not only dependent on the size of the problem in terms of the number of constraints and variables but also on the characteristics of the objective function and constraints.

Constraints can be implemented as either hard constraints or soft constraints. A hard constraint imposes a relation in any condition, while a soft constraint allows small relaxation in the specification of the constraint.

A traditional way to enforce constraints is to translate the constrained problem to a basic unconstrained problem by using a penalty function for constraints. In this way the constrained problem is solved using a sequence of parameterized unconstrained optimizations, which in the limit (of the sequence) converge to the constrained problem. While this method may work for simple constraints, it becomes intractable as the number of constraints grow.

The goal is to integrate the constraints in the fitting algorithm used to estimate the parameters of the primitive building models. For our application it has proven effective to use soft constraints and implement them in the least-squares adjustment from equation (3.21) as weighted observations. The weight specifies the strength of the constraint in the adjustment.

Constraints can be simple constraints or complex constraints that can be described using simple constraints. Parameter constraints are simple constraints expressing a linear relation between two parameters.

$$p_1 + factor * p_2 = offset \tag{3.26}$$

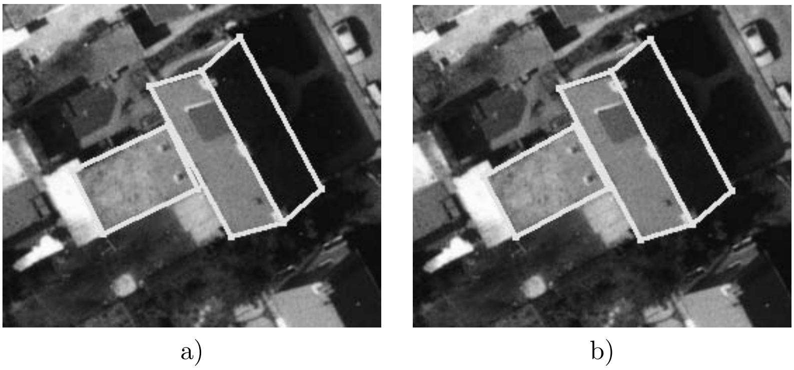
where p_1 and p_2 are the parameters.

A connection constraint can be represented by a parameter constraint specifying that the orientations of the two building primitives are the same ($k_1 = k_2$) and a relation specifying that a corner of one building primitive lies on an edge of the other building primitive. A corner constraint can also be represented by a parameter constraint for the orientations of the two building primitives ($k_1 = k_2$) and a relation that two points expressed in two different coordinate systems are the same. Finally, an extension constraint can be described as a parameter constraint for the orientations of the two building primitives ($k_1 = k_2$), another parameter constraint specifying that either the lengths or the widths of the two building primitives are the same ($w_1 = w_2$ or $l_1 = l_2$) and a corner constraint for a common corner.

Each relation specifying a constraint is included in the least-squares adjustment. Unfortunately, the relations, except the parameter constraints are expressed by nonlinear equations. Therefore, these relations need to be linearized. The linearization can be performed using a Taylor series expansion (see Appendix B).

Fitting a CSG tree to images introduces some modifications to the original fitting algorithm. First the CSG tree is analyzed to determine the constraints between primitives describing the tree. Then the CSG tree is converted into a boundary representation, which is used by the hidden line analysis algorithm. At each iteration of the parameter estimation, the constraints are linearized in the neighborhood of the current estimate and then included together with the observation equations corresponding to the image pixels in the estimation of the parameters of the building.

Figure 3.12 shows the improvement obtained by fitting the whole CSG tree. In the first image, when fitting without constraints the two building models are not connected to each other, there is a slight difference in their orientation. This problem is solved by introducing the connection constraint between the two building models.



a) b)

Figure 3.12: Fitting of a CSG tree. a) Fitting without constraints b) Fitting with constraints

Chapter 4

Evaluation of Building Models

From the set of possible hypotheses of a building model one wants to select the hypothesis that maximizes some appropriate metric. This chapter presents a metric, based on information theory principles, that compares 3D building models to images and chooses the best model.

The concept of mutual information plays a critical role in this thesis. Therefore, the basic mathematics that underly mutual information are introduced in the first sections of this chapter. Different model selection criteria are reviewed. This chapter also contains an analysis and discussion of density estimation techniques with specific focus on Parzen density estimation. Parzen density estimation will play an important role in the computation of mutual information. The final sections of this chapter contain the definition of the metric used for evaluating building models.

Most of the theory presented is cited here without reference. Also, the definition of the concepts is sometimes simplified and the proof of anything that it is easily found in literature is left out. Additional materials can be found in any good book on pattern recognition (e.g. [Duda et al., 2001, Webb, 1999]) and information theory (e.g. [Cover and Thomas, 1991, Hamming, 1980]).

4.1 Information Theory

Before describing the metric for evaluating building models, we first cover some background ideas dealing with information theory. Information theory offers an alternative view of many probability problems, in particular model selection. Coding theory, a part of information theory, concerns the efficient compression of data into a minimal length message. The same ideas can be applied to model selection. The connection is relatively intuitive: a good-fitting model is able to compress its data well.

The information measures for communication systems can be defined through a probabilistic standpoint. Many of the techniques that are common in computer vision can be easily interpreted as statistics of random variables and therefore probability theory can be applied.

4.1.1 Basic Concepts

Shannon is largely credited with founding information theory, originally developed as part of communication theory, with his landmark paper [Shannon, 1948] in 1948. Information theory has been developed for optimizing the transmission of information over a communication channel. In a communication system a message is sent from a source to a destination via a communication channel.

In Shannon's formulation of a discrete communication system, the information source is modelled as a discrete Markov process that generates random symbols from a predefined alphabet. Then the communication channel can be modelled in terms of a probabilistic mapping of input symbols into the set of received symbols in the alphabet of the receiver. Both input and output can be modelled as random variables and the symbols of the input and output alphabet can be seen as events of these random variables.

Consider a set of possible symbols $A = \{a_1, a_2, ..., a_n\}$ whose probabilities of occurrence are $\{P(a_1), P(a_2), ..., P(a_n)\}$, describing a source. This set of symbols can be seen as a discrete random variable, $a_1, a_2, ..., a_n$ are events and $P(a_i))$ is the probability of the event a_i. Clearly,

$$\sum_{i=1}^{n} P(a_i) = 1 \tag{4.1}$$

To measure the information or uncertainty in a communication channel, three conditions should be satisfied. These conditions are:

1. continuity: The functional should be continuous in $P(a_i)$. Small changes in probabilities should give only small changes in the overall information content.
2. extremal property: If all $P(a_i)$ are equal, i.e. $P(a_i) = 1/n$, where n is the number of symbols, then the measure should be monotonically increasing in n. That is, the measure is maximum if all probabilities are equal.
3. additivity: If a choice is broken down into a sequence of choices then the original information should be the weighted sum of the constituent information. That is, if two statistically independent and equally informative messages are received one after the other, then the information gained is twice as much as after receiving the first message. Thus, the information of independent messages can be added.

Shannon proved that the logarithm form was the only functional form satisfying all three conditions.

If the symbol a_i occurs it is deemed to have provided:

$$I(a_i) = \frac{1}{\log_b P(a_i)} = -\log_b P(a_i) \tag{4.2}$$

information. This information is called ***self-information*** and expresses the measure of the uncertainty of the occurrence of the source symbol.

The base of the logarithm b determines the unit of information. If the base is 2 then the unit is bit. If the base is 10 then the information is given in Hartleys. For a natural logarithm the unit of information is a nat. The choice of base 2 is evident when considering digital storage and transmission (bits). Since we are mainly concerned with digital sources and channels, logarithms will be assumed to be base 2 and the subscript will be omitted.

The self-information can be extended to quantify the average information of the source itself. This measure defines the ***entropy*** of the source. The entropy of the source can be calculated as follows. Each symbol a_i occurs with the probability $P(a_i)$ and provides $I(a_i)$ information. The average amount of information obtained per symbol from the source is then:

$$\sum_{i=1}^{n} P(a_i) \cdot I(a_i)$$

Then the average information or entropy, $H(A)$, of the source A is:

$$H(A) = \sum_{i=1}^{n} P(a_i) \cdot I(a_i) = -\sum_{i=1}^{n} P(a_i) \log P(a_i) \tag{4.3}$$

The entropy expresses the uncertainty of the information source or randomness of a random variable. It is a measure of the amount of information required, on average, to describe the random variable. With this definition, it is obvious that the minimum entropy will occur when we are certain with probability 1 as to what the source will do. This can only happen when one of the symbols always occurs with probability 1 while the other symbols occur with probability 0 (i.e. other symbols never occur). In this case there is no uncertainty. Adversely, the entropy will have a maximum value when the uncertainty of what the source will do next is maximum. This occurs if all symbols have equal probability of occurring (i.e. $P(a_i) = \frac{1}{n} \quad \forall i$). An important observation made by Shannon is that any change in the data that tends to equalize the probabilities of the symbols $\{P(a_1), P(a_2), ..., P(a_n)\}$ increases the entropy. Blurring the symbols is one such operation.

Often we are interested in the entropy of pairs of random variables (A, B). Consider an information channel with n input symbols $A = \{a_1, a_2, ..., a_n\}$ and m output symbols $B = \{b_1, b_2, ..., b_m\}$. The probabilities of occurrence of the input symbols are $\{P(a_1), P(a_2), ..., P(a_n)\}$ and the probabilities of the received symbols are $\{P(b_1), P(b_2), ..., P(b_m)\}$.

In the information channel A and B are correlated. If they were independent, it would be impossible to communicate over the channel. The actual symbols transmitted should be the same as those received. However, through the transmission the symbols can be altered by noise. Therefore, the communication channel has to model the noise which can change the received symbol, introduce new symbols or change the information content of what is being transmitted. The information channel can be described by the information channel matrix:

$$\begin{array}{cc} & \textit{output} \\ \textit{input} & \begin{pmatrix} P(b_1|a_1) & P(b_2|a_1) & \cdots & P(b_m|a_1) \\ P(b_1|a_2) & P(b_2|a_2) & \cdots & P(b_m|a_2) \\ \vdots & & & \\ P(b_1|a_n) & P(b_2|a_n) & \cdots & P(b_m|a_n) \end{pmatrix} \end{array}$$

Each row of the information channel matrix corresponds to an input symbol of the channel, and each column corresponds to an output symbol of the channel. When a_i is sent, an output symbol has to be received. Hence

$$\sum_{j=0}^{m} P(b_j|a_i) = 1$$

In case of an ideal channel (i.e. no noise) there is a one to one correspondence between the symbols of the input and the symbols of the output. That is, knowing the symbol that was sent, the received symbol is also known and vice versa. All conditional probabilities are either 0 or 1. Furthermore, each row of the information channel matrix contains only one element with 1 and the rest of the elements are 0.

The ***conditional information*** measuring the surprise of receiving the symbol b_j when it is known that the symbol a_i was sent is:

$$I(a_i|b_j) = -\log P(a_i|b_j) \tag{4.4}$$

The conditional information will be low for high probabilities. For an ideal channel $P(a_i|b_j) = 1$, the conditional information will be 0, so there is no surprise in receiving b_j when a_i was sent.

The uncertainty about the symbol which has been sent when having received b_j is found by taking the expectation over the input alphabet:

$$H(A|b_j) = \sum_{i=1}^{n} P(a_i) \cdot I(a_i|b_j) = -\sum_{i=1}^{n} P(a_i) \log P(a_i|b_j) \tag{4.5}$$

The ***conditional entropy*** is defined as the expectation of the conditional information over the input and output alphabet:

$$H(A|B) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(a_i, b_j) \log P(a_i|b_j) \tag{4.6}$$

The conditional entropy expresses the average information, or uncertainty, of the input alphabet after observing the output of the channel. It measures the average loss of information in the channel. Therefore, $H(A|B)$ can be seen as an indicator of the noise in the channel. For random variables, the conditional entropy reflects how knowledge of one of the random variables reduces the uncertainty of the other. The more A depends on B, the lower the conditional entropy.

Another important concept is that of ***mutual information***. The mutual information between two symbols is defined as the difference between the self-information and the conditional information:

$$I(a_i; b_j) = I(a_i) - I(a_i|b_j) \tag{4.7}$$

This is the amount of information one symbol provides about the other.

An alternative expression for the mutual information can be obtained as follows:

$$\begin{aligned} I(a_i; b_j) &= I(a_i) - I(a_i|b_j) \\ &= -\log P(a_i) + \log P(a_i|b_j) \\ &= \log \frac{P(a_i, b_j)}{P(a_i)P(b_j)} \\ &= I(b_j; a_i) \end{aligned}$$

This expression shows that the mutual information is a symmetric function of the input and the output symbol. That is, a_i tells us as much about b_j as b_j tells us about a_i.

Averaging over all input and output symbols we obtain the ***average mutual information***.

$$I(A;B) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(a_i,b_j)\log\frac{P(a_i,b_j)}{P(a_i)P(b_j)} \tag{4.8}$$

This is the information the channel provides about the input by observing the output. Hence it is called the transinformation. Like mutual information, transinformation is also symmetric $I(A;B) = I(B;A)$. It can be written as a difference between the entropy and the conditional entropy.

$$I(A;B) = H(A) - H(A|B) \tag{4.9}$$

Thus, if the channel is noise-free, we expect $I(A;B) = H(A)$. That is if there is no noise, the amount of information communicated is equal to the uncertainty before communication. In case of a noisy channel we expect the channel to reduce $H(A)$ by the uncertainty $H(A|B)$. If the channel is totally ambiguous, we expect $H(A|B)$ to be no different than $H(A)$ and $I(A;B) = 0$. Hence, the only condition under which the average mutual information of the channel is 0 occurs when the input and output symbols are statistically independent.

The ***joint entropy*** is given by:

$$H(A,B) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(a_i,b_j)\log P(a_i,b_j) \tag{4.10}$$

The joint entropy measures the combined amount of information we have at the input and output of the communication channel. If A and B are totally unrelated, then the joint entropy will be the sum of the entropies of the two alphabets. The more similar (i.e. less independent) the alphabets are, the lower the joint entropy compared with the sum of the individual entropies.

$$H(A,B) \leq H(A) + H(B)$$

The joint entropy, conditional entropy and marginal entropy are related by the chain rule:

$$H(A,B) = H(A) + H(B|A) = H(B) + H(A|B) \tag{4.11}$$

In words, this says that the uncertainty of A and B is the uncertainty of A plus the uncertainty of B given A.

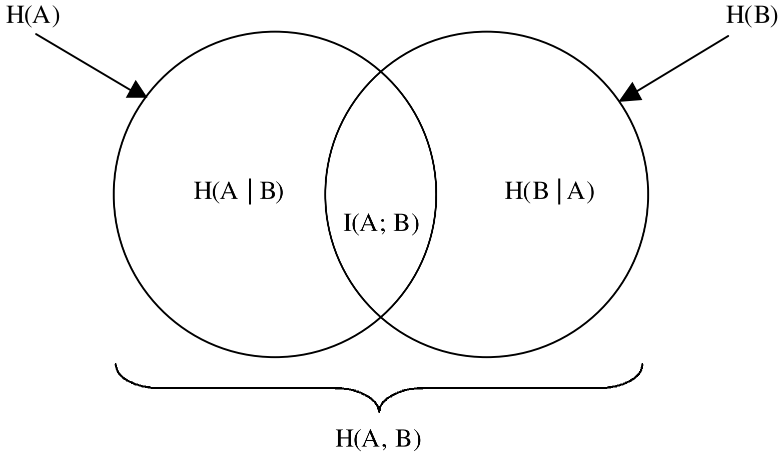


Figure 4.1: Relationship among entropies and mutual information

Combining equation (4.9) and (4.11), we can obtain (Figure 4.1):

$$I(A;B) = H(A) + H(B) - H(A,B) \tag{4.12}$$

The total uncertainty in both A and B $(H(A,B))$ is the sum of the uncertainties in A and B $(H(A) + H(B))$ minus the information provided by the channel.

The ***relative entropy***, also called Kullback-Leiber divergence is a measure of the distance between two distributions. We denote the relative entropy between two distributions $P1$ and $P2$ as $D(P1 \,||\, P2)$, and it is given by:

$$D(P1 \,||\, P2) = \sum P1(x) \log \frac{P1(x)}{P2(x)} \tag{4.13}$$

The definition of the entropy and mutual information for continuous random variables is similar to those of discrete variables. However, the entropy of a continuous source is infinite, since there are an infinite number of possible outcomes. Therefore, the entropy is replaced with another measure called differential entropy.

The differential entropy of a continuous random variable x with the probability density function $p(x)$ is defined by:

$$H(x) = -\int p(x) \log p(x) dx \tag{4.14}$$

Similar to the case of discrete random variables, the average mutual information between two continuous random variables x, y can be defined as follows:

$$I(x;y) = H(x) - H(x|y) \tag{4.15}$$

where

$$H(x|y) = -\int\int p(x,y)\log p(x|y)dxdy \tag{4.16}$$

is the conditional differential entropy of y given x.

Different from the cases for entropy, the properties of mutual information in continuous cases are the same as those in the discrete cases. In particular, the mutual information of the quantized version of a continuous channel will converge to the mutual information of the same continuous channel (as the quantization width goes to zero). Hence, some researchers prefer to define mutual information between two continuous random variables directly as the limit of the quantized channel.

4.1.2 Minimum Description Length

Minimum Description Length (MDL) is a principle of data compression from information theory. It compares the models and data to be represented by the model in terms of the number of bits required to describe both the model and data when encoded in the model.

Suppose we want to communicate a given message through a given communication channel in the shortest amount of time or with the least power. We could send the raw data, but we can reduce the amount of information by encoding the data using a model. Therefore, first raw source data is encoded, then transmitted, and finally decoded.

Table 4.1 gives a concrete example of an encoding scheme in which each letter of an English text is transmitted individually. The encoding scheme in this case consists simply of a lookup-table of code-words that go with each character. This would be a good scheme if each letter has a probability which is time and history invariant. In this case there is a simple algorithm for finding an optimal code, in which the most frequent characters have the shortest code. This technique is called Huffman coding.

The goal is to encode the original data in such way that the number of bits sent is minimum. The number of bits needed to code data gives a measure of the complexity of the data. Consider the example from Figure 4.2. The sender needs to send the model and the residuals. As the residuals are smaller in magnitude than the data, fewer bits are necessary to send the residuals than the original data. If then the sender finds a better model, the residuals will be even smaller, and fewer bits are transmitted. It appears that the more complex the model, the smaller the residuals, and fewer bits need to be transmitted.

Symbol	Probability	Code
A	0.45	0
B	0.13	101
C	0.12	100
D	0.16	111
E	0.09	1101
F	0.05	1100

Table 4.1: Example of Huffman coding

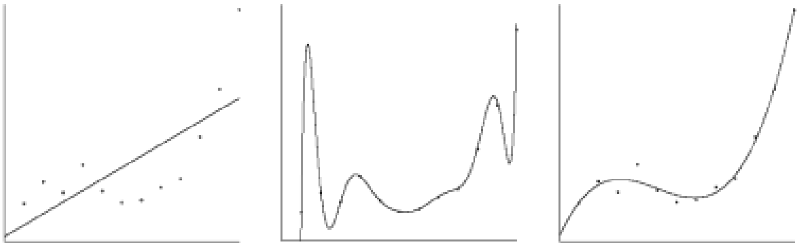


Figure 4.2: Fitting data to models

However, the model needs to be transmitted too and in order to do so the model parameters must be quantified and transmitted. The more sophisticated the model, the greater the compression of the data that can be achieved. But, more bits are required for a complex model which has more parameters. As a result, the gain in description length of the data using a very complex model is more than offset by the large number of bits needed to describe such a model. A very complex model will therefore not lead to the shortest possible description length. On the other hand, using a very simple model, the opposite effect occurs: the description length of the model is very small, but the description length of the data when encoded with the help of the model will be very high. Assuming that we have 'meaningful' data, that is, they do not consist of purely random noise, the shortest possible description length will usually be attained for relatively simple models that fit the data reasonably well. According to MDL, these should be preferred both over very complex models that have no error at all and over simple models that have very high error. In this way, MDL searches for the descriptions which make the best compromise between explaining the data well and providing as simple (general) a model as possible and so minimizing the model description length.

A code C on a set of symbols $A = \{a_1, a_2, ..., a_n\}$ is simply a mapping from A to a set of codewords. In this section, we will consider binary codes so that each codeword is a string of 0s and 1s. A code is called a prefix code if no codeword is a prefix of any other codeword. This means that: all such codes are uniquely decodable and you can decode from left to right. The classic example of a prefix code is Huffman coding (Table 4.1).

Now suppose that symbols of A are generated according to a known distribution P, or in statistical terms, we observe data drawn from P. Given a code C on A with length function L, the expected code length of C with respect to P is defined to be

$$L(C) = \sum_{i=1}^{n} P(a_i)L(a_i) \tag{4.17}$$

where $L(a_i)$ is the length of $C(a_i)$

Our goal is to find a code that has the minimum average length subject to the restriction that the codeword lengths satisfy the fundamental Kraft inequality. This inequality states that any prefix code with codeword lengths $L(a_1), L(a_2), ..., L(a_n)$ must satisfy

$$\sum_i 2^{-L(a_i)} \leq 1 \tag{4.18}$$

Therefore, given a prefix code C on A with length function L we can define a distribution on A as follows:

$$Q(a_i) = \frac{2^{-L(a_i)}}{\sum_j 2^{-L(a_j)}} \tag{4.19}$$

Conversely, for any distribution Q on A, we can find a prefix code with the length $L(a_i) = \lceil \log Q(a_i) \rceil$.

To quantify the length of a coded message, information theorists use the concept of entropy. Shannon's source coding theorem states that the expected length $L(C)$ of any prefix code satisfies:

$$L(C) \geq H(P) \tag{4.20}$$

where $H(P)$ is the entropy of the source. The equality holds if and only if $L = -\log P$. That is, the optimal code length is equal to the self-information.

This theorem states that the entropy provides a lower bound on code length. By applying Huffman's algorithm to the distribution P on A, we obtain a code that is nearly optimal in expected code length. Cover and Thomas ([Cover and Thomas, 1991]) proved that the Huffman code for A has an expected length no greater than $H(A) + 1$.

It must be emphasized however, that any distribution Q defined on A, but not necessarily the data-generating or true distribution P, can be used to encode data from A. Actually, in most applications, the true distribution P is rarely known. In these cases if we design a code based on a distribution Q, we get:

$$\begin{array}{ll} L(C_{suboptimal}) & = \sum_i P(a_i) - \log Q(a_i) \\ & = \sum_i P(a_i) \log(P(a_i)/Q(a_i)) - \sum_i P(a_i) \log Q(a_i) \\ & = D(P \,||\, Q) + H(P) \end{array} \tag{4.21}$$

So the relative entropy is the extra price you pay in terms of average code length you pay for using an incorrect probability distribution. If we knew the true probability distribution P, then we could construct a code using this distribution and it would take a minimum of $H(Q) = H(P)$ bits on average to represent a symbol. If, however, we used a different distribution Q, then we pay a penalty for not using the correct distribution. This penalty is relative entropy and it is the average number of extra bits that it will cost you to use distribution Q when the real distribution is P. Instead of $H(P)$ bits, you will need $H(P) + D(P \,||\, Q)$ bits.

Alternatively, the shortest possible average length for any prefix code for a sample sequence of values of any random variable is the entropy of that variable. Furthermore, given a sufficiently long sequence of source values to encode, there exists a code which approaches this limit arbitrarily closely.

Ultimately, the crucial aspect of the MDL framework is not found in the specifics of a given coding algorithm, but rather in the code length interpretation of probability distributions. For simplicity, we will refer to $L = -\log P$ as the code length of (the code corresponding to) a distribution P, whether or not it is an integer.

4.2 Model Selection

If more models are possible, the fitting procedure needs to consider all the potential models and select which of these models fits the image data best. This is the task of model selection. So, the model selection problem can be formulated as: given the data D and a set of models $M = \{M_1, M_2 ...\}$, find the model which best describes the data. This section reviews current statistical methods in model selection.

In order to select the best model among a number of possible candidates we need a criterion for the "goodness of a model". The straightforward way would be to use residuals, which measure the goodness of the fit. Unfortunately, the residual cannot serve this purpose, since models with more parameters might fit the data better than models with less parameters. As outlined in section 4.1.2, such a model is not what we want.

We must find a suitable compromise for the complexity of the model, this is often referred to as the principle of parsimony. If the model is too complex it will capture the errors in the data. If it is too simple it will not be a sufficient estimate of the process that created the data. Ideally, the model should be chosen so that it encodes the meaningful aspects of the data and discards the unstructured portion.

Several model selection criteria have been introduced over the years. They are usually based on the following principles: hypothesis testing, bayesian rule, information criterion, MDL principle and empirical criteria [Kanatani, 2000]. Empirical criteria include cross-validation, jackknife, and bootstrap [Efron and Tibshirani, 1993]. A good review of model selection criteria with focus on two view geometry determination is given in [Torr, 1999]. In [Bubna and Stewart, 2000] different model selection techniques used in range data segmentation algorithms are presented.

4.2.1 Hypothesis Testing

One method of model comparison is via hypothesis testing. Different models are usually sorted by the complexity from the most complicated model to the most simple. To perform a model comparison via hypothesis testing we have to test the null hypothesis that one model M_1 fits the data, by comparing it to an alternate hypothesis that the model M_2 fits the data. A common way to do this is by likelihood ratio test. To do this the maximum likelihood estimation of the parameters θ_1 of the model M_1 and θ_2 of the model M_2 must be computed. The following test statistics is examined:

$$\lambda(D) = log\left(\frac{P(D|M_1, \theta_1)}{P(D|M_2, \theta_2)}\right) = L_1 - L_2 \tag{4.22}$$

where L_i represents the likelihood and the model M_1 has more parameters than M_2. This statistic asymptotically follows a χ^2 distribution with $p_1 - p_2$ degrees of freedom, where p_i is the number of parameters of the model M_i.

Then the following test is performed:

$$\lambda(D) = L_1 - L_2 < \chi^2(\alpha, p_1 - p_2) \tag{4.23}$$

where α is a significance level. If $\lambda(D)$ is less than some threshold then the model M_2 is accepted, otherwise the model M_2 is rejected and the next model is tested, and so on until a model is accepted.

There are some problems with this approach. First, the hypothesis testing procedure requires a structural hierarchy between the models. So it is only suitable for nested models, i.e. the parameters of one model form a subset of the parameters of a more general model. Furthermore, the hypothesis testing is difficult to adapt to the situation where several models might be appropriate, because it is difficult the choose the significance level and the threshold for the test.

4.2.2 Akaide Information Criterion

A good criterion to evaluate which model is better than the other should take into consideration both the goodness of fit and the complexity of the model. Hence, the complexity should be penalized and the score function should look like:

$$Score(M) = error(M) + penalty\_function(M) \tag{4.24}$$

Akaike was the first who tried to tackle this problem [Akaike, 1974]. He derived a formula that predicts the residual of the data that could be captured in the future. This formula is usually referred to as Akaike Information Criterion (AIC). To be able to predict future residuals he had to make assumptions about the noise in the data. He proved that in regression problems the number of parameters increases the expected residuals of future data.

AIC has the form:

$$AIC = -2L + 2p \tag{4.25}$$

where p is the number of parameters of the model and L is log-likelihood. AIC has two terms, the first one corresponds to the badness of fit and the second one is a penalty for model complexity. The best model is that which has minimum value of AIC. So, the best model is the one with highest information content but least complexity.

The advantage of AIC is its simplicity. It is very easy to calculate once the maximum likelihood estimate of the model parameters is done. However, AIC introduces only a constant penalty per parameter. If the number observations increases the wrong model will be chosen. Even Akaike states that he thinks AIC is only useful when Bayesian analysis is impractical.

4.2.3 Bayesian Model Selection

Criteria based on Bayes rule compute the posterior probability of each model directly and select the one with the highest posterior. A set of models $M = \{M_1, M_2, ..., M_m\}$ that can describe the data D is given. Suppose we can associate with each model M_i a prior probability $P(M_i)$ according to the background knowledge about the models. These probabilities sum to one. Actually $P(M_i)$ is the initial degree of belief that M_i is the correct model. We assume that the conditional probability $P(D|M_i)$ that the data D would be experienced, given that M_i is the correct model, can, in principle, be computed.

Bayes rule states:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{P(D)} \tag{4.26}$$

The posterior probability that a given model M_i is the correct one is:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{\sum_{j=1}^{m} P(D|M_j)P(M_j)} \tag{4.27}$$

The marginal probability $P(D|M_i)$ is obtained by integrating out over all possible values of the parameters of the model:

$$P(D|M_i) = \int P(D|M_i, \theta_i)P(\theta_i|M_i)d\theta_i \tag{4.28}$$

For comparing two models M_i and M_j, the posterior odds can be used:

$$O_{ij} = \frac{P(M_i|D)}{P(M_j|D)} = \frac{P(D|M_i)P(M_i)}{P(D|M_j)P(M_j)} \tag{4.29}$$

The term

$$B_{ij} = \frac{P(D|M_i)}{P(D|M_j)} \tag{4.30}$$

is called Bayes factor.

As it can be seen from equation (4.29), the Bayesian model selection approach depends on the a priori probabilities $P(M_i)$. In the absence of any information about the models, all the models can be assumed equally likely, i.e. their a priori probabilities can be considered equal $P(M_i) = P(M_j)$. This leads to the Maximum Likelihood (ML) criterion that chooses the model which has maximum likelihood $P(D|M_i)$ given the data D.

The MAP (Maximum A Posterior) method is the true Bayesian approach, since the ML technique does not even make use of Bayes rule. The MAP approach is more complete, but ML is often used when one does not want to work with prior probabilities for some reason (perhaps the prior probabilities are unknown). The approaches coincide when the prior distributions weight each model equally.

Bayesian Information Criterion

In order to compute the Bayes factor, the a priori distributions $P(\theta_i|M_i)$ of each model have to be computed. This allows incorporation of a priori information, if this information is available. But if this information is not available then usually it is hard to compute.

The integral from equation (4.28) may be difficult to compute. One way to calculate it is by means of analytical approximations. The simplest one was introduced by Schwarz and it is commonly known as Bayesian Information Criterion (BIC)

[Schwarz, 1978]. This approximation assumes that $P(\theta_i)$ is approximately normal. It has the following form:

$$BIC = -2L + p \log N \tag{4.31}$$

where N is the number of observations. The model with the lowest BIC is chosen as the best one.

4.2.4 Minimum Description Length Model Selection

In the section 4.1.2 the use of code length interpretation of probability distributions was motivated and the use of models for building good codes was illustrated. Probability models or descriptions for each binary string were evaluated on the basis of their code length. One wants to find a concise model that fits the important features of the data. A concise model should be easy to describe, while a good fit implies that the model makes easy the description of the data. Rissanen ([Rissanen, 1978]) formulates such thinking in his Principle of Minimum Description Length:

Choose the model that gives the shortest description of data.

The MDL principle selects the model M_i with the shortest complete description of the data. When the data is encoded using a model, then the model as well as the description of the data with respect to that model has to be considered, that is:

$$L(M_i, D) = L(M_i) + L(D|M_i) \tag{4.32}$$

where $L(x)$ is the description length of x and $L(M_i, D)$ has to be minimized.

If the code used for the description is optimal, then the length of the description is equivalent to its information content. Conform to Shannon's source coding theorem (4.20), the length of the description is:

$$L(x) = -\log P(x) = I(x) \tag{4.33}$$

Thus $L(M_i) = I(M_i)$ and $L(D|M_i) = I(D|M_i)$

Then the description length (4.32) becomes :

$$L(M_i, D) = I(M_i) + I(D|M_i) \tag{4.34}$$

By using the definition of the mutual information (4.7):

$$I(D|M_i) = I(D) - I(D; M_i) \tag{4.35}$$

the formula of description length from (4.34) can be expressed as:

$$L(M_i, D) = I(M_i) + I(D) - I(D; M_i) \tag{4.36}$$

Since $I(D)$ is constant, in order to minimize the description length $L(M_i, D)$ the expression $I(D; M_i) - I(M_i)$ has to be maximized.

Therefore, it follows that the best model is given by:

$$M_{opt} : \max_i (I(D; M_i) - L(M_i)) \tag{4.37}$$

MDL has a connection with Bayesian approaches of model selection. Actually, it can be shown that the MAP and the MDL principles lead to the same solution.

Since $P(D)$ is constant, by using Bayes rule (4.26) MAP states that $P(D|M_i)P(M_i)$ has to be maximized.

The description length from equation (4.34) can be written as:

$$\begin{aligned} L(M_i, D) &= I(M_i) + I(D|M_i) = -\log P(M_i) - \log(D|M_i) \\ &= -\log P(D|M_i)P(M_i) \end{aligned} \tag{4.38}$$

Minimizing the description length is equivalent to maximizing $P(D|M_i)P(M_i)$, which is actually the MAP criterion.

4.2.5 Empirical Model Selection

Several approaches for model selection have been introduced that require additional data for evaluation. The empirical model selection methods, such as cross-validation, jackknife and bootstrap [Efron and Tibshirani, 1993] are based on resampling the data that are being fit. A large data set is needed for obtaining a stable fit and also for accurately validating the model.

In case of the cross-validation, the data is divided in two sets, a learning and a validation set. The models are fit to the learning set and their residuals are evaluated on the validation set. A complex model may fit very well to the learning set, but should result in a large deviation for the validation set. Cross-validation can be used to estimate the generalization error of a given model, therefore, it can be used in model selection for choosing the one of several models that has the smallest estimated generalized error. When the size of the validation set is 1, the method is called leave-one-out cross-validation. In this work, the cross-validation is used at the estimation of the probability distributions and it is described in more detail in the next section of the thesis.

The jackknife method is similar to leave-one-out cross-validation. Both involve omitting one sample in turn and refitting the model on the remaining data. But, cross-validation is used to estimate the generalization error, while the jackknife is used to estimate the bias. Several iterations are performed and the models are evaluated using the average of the residuals of the randomly selected sample.

In the bootstrap method, the testing set is generated via a computer simulation by simulating the mechanism according to which the data have been produced. In the simplest form of the bootstrapping, instead of repeatedly analyzing subsets of the data, subsamples of the data are repeatedly analyzed. Each subsample is a random sample with replacement from the full sample.

A lot of theoretical studies have been done on equivalence relations and interrelationships among different model selection criteria. These studies revealed that all criteria behave more or less similarly and particularly so asymptotically. The conclusion is that one may choose the criterion that is easiest to formulate or compute for a given problem. In this project MDL was used, because the mutual information based evaluation function can be easily integrated into an MDL framework.

4.3 Density Estimation

The techniques for selecting the model which best describes the data assume that the a priori and conditional probabilities of the data given the model are known. Therefore, in this section we will describe a number of techniques for estimating densities from samples.

Density estimation is the problem of modelling a density $P(x)$ given a finite number of samples x_n drawn from that density function. For this purpose we will have a finite number of samples. There are two basic approaches to perform density estimation:

- parametric: a given form for the density function is assumed (i.e., Gaussian) and the parameters of the function (i. e., mean and variance) are then optimized by fitting the model to the data set. Usually the parameters are estimated using the Maximum Likelihood approach.
- non-parametric: no functional form for the density function is assumed, and the density estimate is driven entirely by the data. Non-parametric techniques are histogramming, k Nearest Neighbor and Parzen window density estimation.

In general, a parametric density estimation is a good choice if the assumed form is a reasonable approximation. In this case, fewer training data are needed and computing the density is fast. For non-parametric density estimation no search for parameters is needed. While parametric methods use the parameters as the model, non-parametric methods use the sample to directly define the model. The non-parametric schemes on which we will focus are histogramming and Parzen window density estimation.

4.3.1 Non-parametric Density Estimation

The probability that a random variable x, drawn from a distribution $P(x)$, will fall in a region $\mathcal{R}$ of the sample space is

$$P = \int_{\mathcal{R}} P(x')dx' \tag{4.39}$$

Suppose now that N vectors $x_1, x_2, ..., x_N$ are drawn from the distribution. The probability that k of these N vectors fall in $\mathcal{R}$ is given by the binomial distribution:

$$P(k) = \left(\begin{array}{c} N \\ k \end{array} \right) P^k(1-P)^{(N-k)} \tag{4.40}$$

The expected number of samples falling in $\mathcal{R}$ is the expected value, which in the case of a binomial distribution is:

$$E[k] = NP \quad \text{and} \quad E\left[\frac{k}{N}\right] = P \tag{4.41}$$

and the variance is:

$$var\left[\frac{k}{N}\right] = E\left[\left(\frac{k}{N} - P\right)^2\right] = \frac{P(1-P)}{N} \tag{4.42}$$

Therefore, as $N \to \infty$, the variance decreases, so a good estimate of the probability P can be obtained from the mean fraction of points that fall within $\mathcal{R}$.

$$P \approx \frac{k}{N} \tag{4.43}$$

On the other hand, if $\mathcal{R}$ is so small that $P(x)$ does not vary appreciably within it, then

$$\int_{\mathcal{R}} P(x')dx' = P(x)V \tag{4.44}$$

where V is the volume enclosed by the region $\mathcal{R}$.

Merging (4.43) and (4.44) we obtain:

$$P(x) \approx \frac{k}{NV} \tag{4.45}$$

This estimate becomes more accurate as we increase the number of sample points N and shrink the volume V.

4.3.2 Histogramming

The simplest form of non-parametric density estimation based on equation (4.45) is the traditional histogram. It divides the sample space into a number of bins and approximates the density at the center of each bin by the fraction of points in the training data that fall into the corresponding bin.

$$p_n(x) = \frac{k_n}{NV_n} \tag{4.46}$$

where N is the total number of samples, k_n is the number of samples falling in bin n, and V_n is the volume of this bin. The histogram requires two parameters to be defined: the bin width and the starting position of the first bin.

The histogram is a very simple form of density estimation, but it has various drawbacks. The final shape of the density estimate depends on the starting point of the bins. For multivariate data, the final shape of the density is also affected by the orientation of the bins. The discontinuities of the estimate are not due to the underlying density, they are only artifacts of the chosen bin locations. These discontinuities make it very difficult to grasp the structure of the data. A much more serious problem is the curse of dimensionality, since the number of bins grows exponentially with the number of dimensions. In high dimensions we would require a very large number of examples or else most of the bins would be empty.

The variance of the density estimation is:

$$var(\hat{p}(x)) = \frac{var(k)}{NV^2} = \frac{p(1-p)}{NV^2} \tag{4.47}$$

and the bias is:

$$bias(\hat{p}(x)) = E(\hat{p}(x) - p(x)) = \frac{1}{NV}E(k) - p(x) = \frac{p}{V} - p(x) \tag{4.48}$$

The bins largely determine the accuracy of the histogram. If the bins are too wide the variance of the histogram is low, but the bias is high. If the bins are too narrow the variance increases, but the bias is small. At the limit, when $V = 0$, the histogram is precisely the empirical probability density function, with infinite variance. Therefore, it is important to find the bin size that balances the variance and bias.

The estimate becomes more accurate as we increase the number of sample points N and shrink the volume V_n (Figure 4.3). But, in practice the value of N is fixed (the total number of samples). In order to improve the accuracy of the estimate $p_n(x)$ we could let V_n approach to zero but then the region would become so small that it would enclose no samples. This means that a compromise value of the volume V_n has to be found. It should be large enough to include enough samples within a region and small enough to support the assumption that $p_n(x)$ is constant within region.

There are three condition that have to be fulfilled by an estimator in order for this to converge to $p(x)$:

1. $lim_{N \to \infty} V_n = 0$ guarantees that local properties are estimated.
2. $lim_{N \to \infty} k_n = \infty$ assures sufficient points in the neighborhood.
3. $lim_{N \to \infty} \frac{k_n}{N} = 0$ guarantees that $\frac{k_n}{NV_n}$ converges.

It is a fact that, as the data set gets larger and the histogram bins get smaller, the histogram divided by the total number of data items will almost certainly converge to the probability density function.

Applying this to practical density estimation problems there are two basic approaches that can be adopted in order to meet these conditions. One strategy is to choose a fixed value of k_n (e.g. $k_n = \sqrt{N}$) and determine the corresponding volume V_n from the data. Actually, the volume V_n is increased until it encloses k_n neighbors of x. This gives rise to the k Nearest Neighbor (kNN) approach. A second strategy is to choose a fixed value of the volume V_n and determine k_n from the data. This leads to methods commonly referred to as Kernel Density Estimation (Parzen window density estimation). It can be shown that both kNN and Parzen window density estimation converge to the true probability density as $N \to \infty$, provided that V_n shrinks with N, and k_n grows with N.

4.3.3 Parzen Window Density Estimation

The Parzen window approach for estimating densities can be introduced by assuming the regions R_n to be d-dimensional hypercubes with edge-dimension h_n. Let $V_n = (h_n)^d$ be the volume of the hypercubes centered at x. For N sample points x_n, the density takes the following form:

$$p_n(x) = \frac{1}{Nh^d} \sum_{n=1}^{N} K\left(\frac{x - x_n}{h}\right) \tag{4.49}$$

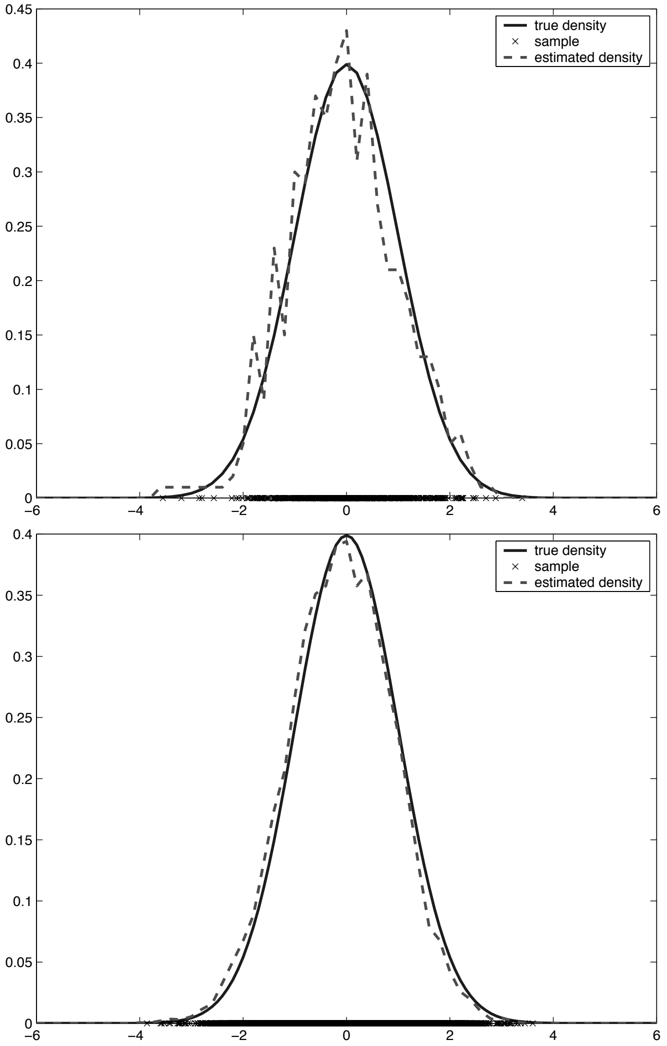


Figure 4.3: Gaussian density with mean 0 and variance 1 and the histogram density estimate computed from the sample. Top a sample of $N = 100$ points were drawn from the density. Bottom $N = 1000$ points.

The function K is often called the kernel or window function. The quality of the approximation is dependent both on the functional form of K, and its width. Different window functions will lead to very different density estimates.

If the kernel function is:

$$K(u) = \begin{cases} 1 & \text{if } |u_j| < 1/2 \quad j = 1...d \\ 0 & \text{otherwise} \end{cases} \tag{4.50}$$

then the Parzen window estimate resembles the histogram, except that the bin locations are determined by the data points. In this case the density estimate still has discontinuities and it weights equally all points x_n regardless of their distance to the estimation point x.

These drawbacks can be overcome with using a smooth kernel function $K(u)$ which satisfies the condition:

$$\int_R K(x)dx = 1 \tag{4.51}$$

Intuitively, the Parzen density estimator computes a local, or windowed, average of the sample. If K is symmetrical about the origin we can view the window function as being centered on the query point x, rather than at the data points. Viewed in this light, the density estimate at a query point is a weighted sum over the sample, where the weighting is determined by the window function. The most common window functions are unimodal, symmetric about the origin, and fall quickly to zero. In effect, the window function defines a region centered on x in which sample points contribute to the density estimate. Points that fall outside of this window do not contribute.

The Gaussian density is a common selection for K, making the Parzen density estimate a mixture of Gaussians. There is one Gaussian centered at each sample.

$$K(x) = \frac{1}{(2\pi\Sigma)^{d/2}} \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right) \tag{4.52}$$

where Σ is the variance of the Gaussian.

Figure 4.4 contains a graph of a density, a sample of $N = 100$ points, and the Parzen estimate constructed from the sample. The Parzen density estimate shows better approximation of the true density than the histogram estimation (Figure 4.3 top) for the same number of sample points.

The density estimate at x is the ratio of the number of weighted sample points within the window divided by the total number of sample points, N. Getting a reliable estimate of this ratio involves having a reasonable number of points falling into the window around the query point. The number of points that we expect to fall into this window is a function of both the size of the sample and the size of the

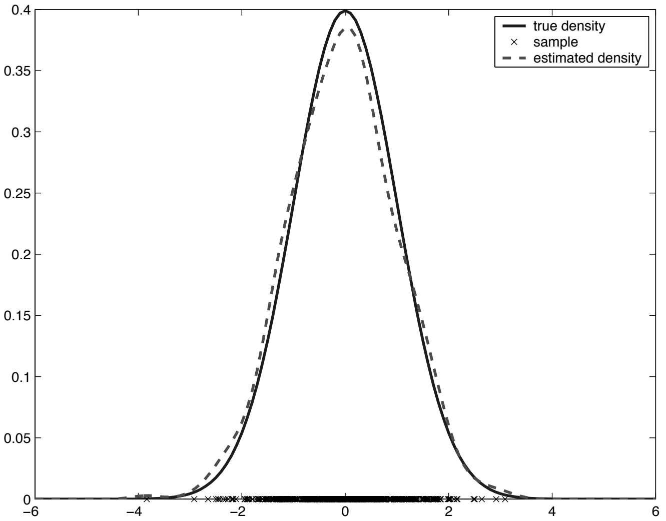


Figure 4.4: Gaussian density with mean 0 and variance 1 and the Parzen density estimate with $\sigma = 0.3$ computed from a sample of $N = 100$ points.

window. As the number of points that fall into a window decreases, the variance of the Parzen density estimate increases.

It was shown in [Duda et al., 2001] that the p_n converges to p by mean square convergence, that is:

$$\lim_{n \to \infty} p_n(x) = p(x) \tag{4.53}$$

and

$$\lim_{n \to \infty} \sigma^2[p_n(x)] = 0 \tag{4.54}$$

Choosing the Width of Window

The parameter h in (4.49) is also called the smoothing parameter or bandwidth. The problem of choosing the width of the window is crucial in density estimation. Figure 4.5 contains a graph of four Parzen density estimates with different bandwidths. A large bandwidth will oversmooth the density and mask the structure of the data, while a small bandwidth will yield a density estimate that is spiky and will be a noisy version of the true $p(x)$.

If we had access to an unlimited number of samples the solution is straightforward: choose the window width that produces the lowest error rate. In real applications

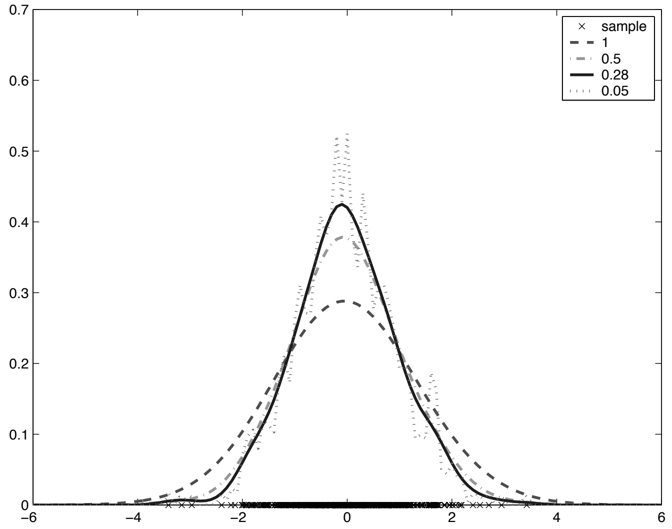


Figure 4.5: Parzen window density estimates with different smoothing parameters

we only have access to a finite number of samples. Generally, more sample data leads to a better estimation. However, estimating the density with very large data sets can be difficult, and there are diminishing returns.

One approach is to use the entire data set to select the window width. This approach has two major problems. The final density estimate will normally overfit the sample data set and it will not be able to generalize to this new data. The error rate estimate computed on the sample data set will be lower than the true error rate.

A much better approach is to split the sample data set in two disjoint subsets. This method is known as *holdout*. The estimator is trained on one subset and tested on the other. This is a waste of data, particularly if we have small number of samples. Since the holdout method is a single train-and-test experiment, the error rate can be misleading, if we happen to get an "unfortunate" split.

The limitations of the holdout method can be overcome with resampling methods at the expense of a higher computational cost. These resampling methods include cross-validation and bootstrapping.

Instead of performing one split of the data set as for holdout, the *cross-validation* method performs more splits of data set. In particular, the density could then be estimated by averaging over all possible splits. The most usual form of this algorithm involves omitting single items from the data set, and is known as *leave-*

one-out cross-validation.

The *bootstrapping* is a resampling technique with replacement. For a data set with N samples, randomly select with replacement N samples and use this set for estimating the density. The remaining samples that were not selected for estimating the density are used for testing. This process is repeated a specified number of times.

4.3.4 Comparison

In comparison between the histogramming and the Parzen window methods are actually quite similar. In fact, if the kernel is set to a unit square pulse, the Parzen window density estimator behaves much like the traditional histogramming method. So, the Parzen window density estimation includes histogramming.

If we compare the two methods on speed, we see that histogramming is fast, but the Parzen windowing provides a smoother solution to the problem. At the bin boundaries the histogram will deliver a discontinuous probability. The Parzen estimation gives a density that has infinitely many derivatives. Even with the smoothness, Parzen windowing can be prone to error. In addition to providing a smooth estimate, the Parzen window method assumes it is approximating a smooth density. This assumption may not be true. In this case the Parzen estimate will not converge asymptotically to an accurate estimate.

Which method is best? There is no clear answer. For accuracy, Parzen windowing provides a more continuous density estimate, more closely hugging the real density function if the real density is continuous. The downside is that the Parzen window method uses greater computational resources.

4.4 Metric for Evaluating Building Models

The 3D reconstruction of a building can be seen as a tree search (Figure 1.2). In this approach, the search space for the best fitting building model can be represented as a tree with the nodes of the tree representing the different building primitive hypotheses. The root node of the tree represents the initial state, where only the ground plan of the building is known. The first level of the tree contains all the possible partitioning schemes of a building. The second level contains the partitions corresponding to each partitioning scheme. The last level of the search tree contains the different building hypotheses generated for each building partition. The tree can have an extra level containing different height values that have to be checked in case of a terrain with large height variations. Note that this tree is not a search tree in a traditional sense, since decisions do not have to be made at each level. At the third level, containing the partitions forming a partitioning scheme, each partition has to be evaluated, but no best partition is selected.

A problem that has to be considered at the search of the tree is the definition of a score function to guide the search to obtain the best solution. In other words we need a metric that can compare different building hypotheses and select the best one.

In many applications, such as object recognition, stereo vision, 3D reconstruction, and tracking, there is a need to evaluate the matching of object models and image data. From the possible hypotheses of matching between an object model and an image, one wants to select the hypothesis that maximizes some appropriate metrics. Therefore, an evaluation function has to be defined to measure the quality of the match. Usually, an evaluation function is based on error models that describe how an image feature may differ from what the object model has predicted. Two main categories of approaches for defining evaluation functions can be distinguished. Simple evaluation functions were used in [Ayache and Faugeras, 1986], [Beveridge et al., 1989], [Grimson, 1990]. With this approach, components of the evaluation function are combined using trade-off parameters that are determined empirically. Another class of evaluation functions is based on statistical theory. Match quality measures are often defined using Bayesian probability theory ([Pope and Lowe, 1994], [Wells, 1993]).

Given an object model (shape) and a pose (coordinate transformation), a model for the imaging process could be used to predict the resulting image. The predicted image could then be compared to the actual image directly. If the object model and pose are correct the predicted and actual images should be identical, or close to it. But, the relationship between an object model and the object's image is rather complex. The appearance of the object model in the image depends on a lot of factors, such as the surface properties, the orientation of the object, the lightning and the viewing direction. As an alternative, the object models and images are represented as collections of edges and a distance metric is defined between them. For example Huttenlocher [Huttenlocher et al., 1991] use the Hausdorff distance. There are many methods that use a metric that is proportional to the number of edges that coincide. A version of such a metric can be defined by introducing a penalty for unmatched edges ([Lowe, 1987], [Wells, 1993]). The drawback of these methods is that they depend on edge extraction.

To overcome these problems we propose a metric that compares 3D object models directly to raw images. No preprocessing or edge detection is required. The metric is based on a formulation of the mutual information between the object model and the images. In computer vision, mutual information has been used for relational matching ([Vosselman, 1992]) and for medical image registration ([Viola and Wells, 1997], [Collignon et al., 1995]) among others.

The mutual information matching criterion is based on the assumption that the statistical dependence between a model and image is maximal if the model is correct. Our metric combines two measures. One of them counts evidence along the model contour in the images. If the hypothesis is correct, we expect to find high image gradients along the projected model contours. The second one measures image intensity similarity over two images. Given an object hypothesis and a pose, a point-to-point mapping can be defined between images. If the hypothesis is correct then the intensities at corresponding pixels will be highly correlated.

The technique used for comparing 3D object models to images does not require a priori information about the surface properties of the object and is robust with respect to variations of illumination. As result the method is quite general and may be used in a wide variety of applications.

4.4.1 Mutual Information of Contours

A 3D object model can be projected in the image if its pose is known. This projection of the object can be matched against the image data. The quality of this match can be measured by the mutual information between the model contour and image data. If the model is correct, then high image gradients are expected along the projected model contours.

In an information theoretic approach, matching can be seen as a communication problem, where the model description $M = \{m_1, m_2, ...\}$ is transmitted through a communication channel into the image $D = \{d_1, d_2, ...\}$. The image data will be similar to the model data but sometimes corrupted due to occlusions, noise, etc. The similarity between the two descriptions can be measured by the mutual information $I(M; D)$.

The mutual information between the model description and the data description is defined in (4.7) as:

$$I(m_i; d_j) = I(d_j) - I(d_j|m_i) \tag{4.55}$$

where $I(d_j) = -\log P(d_j)$ and $I(d_j|m_i) = -\log P(d_j|m_i)$

Thus the mutual information can be written as:

$$I(m_i; d_j) = \log \frac{P(d_j|m_i)}{P(d_j)} \tag{4.56}$$

or

$$I(m_i; d_j) = \log \frac{P(d_j, m_i)}{P(d_j)P(m_i)} \tag{4.57}$$

The description of the model and image data used in the definition of the mutual information depends on the level of abstraction chosen. We could use features to describe both model and images. Then it would be necessary to segment the images before the matching and usually the segmentation needs selection of a threshold. Also the extracted features are influenced by noise, bad contrast and occlusions in the image. To overcome these problems we do the matching between the model and the images and the evaluation of the matching at the lowest level of abstraction, namely at pixel level. The attributes dealt with at this level are gradients.

Independent Image Pixel Model

A simple description considering an image as a set of isolated pixels allows us to apply the concepts of information theory directly. We confer to an image pixel $point_i$ a random variable $grad_i$ representing the magnitude of the gradient computed from the image in that pixel. The probability distribution of the random variable $grad_i$ is $P(grad_i)$ and can be computed from the image.

Considering an image pixel as an information source, we can define the information produced by a gradient value $grad_i$ by:

$$I(point_i) = -\log P(grad_i) \tag{4.58}$$

This measures the surprise of occurrence of a gradient value.

Now consider two information sources: a model and an image. If a model point $point_m$ is the input for an information channel, and the gradient $grad_i$ of an image pixel $point_i$ corresponding to the model point is the output, then the mutual information between the image pixel $point_i$ and the corresponding model point $point_m$ using (4.56) is given by:

$$I(point_m; point_i) = \log \frac{P(grad_i|point_m)}{P(grad_i)} \tag{4.59}$$

This quantity measures the reduction in the surprise of the input $point_i$ due to the fact that the given image point corresponds to a model point. The distribution of the gradients at random image points $P(grad_i)$ and also the conditional distribution of the gradients along the projected model contour $P(grad_i|point_m)$ can be determined by training.

Our evaluation function gives a positive response where points match with high confidence, a negative response where there is a clear mismatch, and a zero response at points where there is neither evidence for, nor evidence against, a match.

Then assuming independence between image pixels, the mutual information between a model line and an image line is found by taking the sum of the mutual information of the points of the line:

$$I(line_m; line_i) = \sum_{point_m \in line_m} I(point_m; point_i) \tag{4.60}$$

If more images are available, then mutual information for a model is given by the sum of the mutual information over all points on all projected model lines in all images:

$$I_{contour}(M; D) = \sum_{k=1}^{\#img} \sum_{line_m} \sum_{point_m} \log \frac{P(grad_i|point_m)}{P(grad_i)} \tag{4.61}$$

Markov Random Field Image Model

In a general case we have to take into consideration the neighborhood of a pixel. The dependencies introduced by the neighborhood of a pixel can be modelled by a Markov Random Field (MRF). In the framework of MRF, a set of sites $S = \{s_1, s_2, ..., s_N\}$ are given. The sites in S are related one to another through a neighborhood system: $\mathcal{N} = \{\mathcal{N}_i | \forall s \in S\}$. The sites have labels $\Lambda = \{g_1, g_2, ..., g_M\}$. A MRF on these sites is defined by the graph $G = (S, \mathcal{N})$ [Li, 2001].

In [Hamming, 1980] Hamming defined the information of a Markov process. For an m-th order stochastic process $\{t_{i1}, t_{i2}, ..., t_{im}, t_i\}$, the amount of information obtained when a symbol t_i is received is:

$$I(t_i|t_{i1}, t_{i2}, ..., t_{im}) = \log \frac{1}{p(t_i|t_{i1}, t_{i2}, ..., t_{im})} \tag{4.62}$$

Extending Hamming's definition to MRF, we get the information produced by the gradient of a site having known the gradients of other sites:

$$I(s_i|s_j, i \neq j) = \log \frac{1}{P(s_i|s_j, i \neq j)} = \log \frac{1}{P(s_i|s_j, j \in \mathcal{N}_s)} \tag{4.63}$$

Then, the mutual information between an image pixel and the corresponding model point is given by:

$$I(point_m; point_i|point_j, j \neq i) = \log \frac{P(grad_i|point_m, grad_j, point_j \in \mathcal{N}(point_i))}{P(grad_i|grad_j, point_j \in \mathcal{N}(point_i))} \tag{4.64}$$

The estimation of the probabilities from equation (4.64) is very difficult, since even in case of the four nearest neighbors and $M = 256$ labels, we have 256^5 possible realizations of $(point_i, point_{i1}, point_{i2}, point_{i3}, point_{i4})$. Therefore, we have to reduce the number of possible values.

According to Hammersley-Clifford theory, the conditional probability of a label of a site with respect to its neighbors can be expressed as a Gibbs distribution:

$$P(s_i|s_j, j \in \mathcal{N}_i) = \frac{1}{Z_i} \exp\left(-U_i(s)\right) \tag{4.65}$$

with the local energy given by:

$$U_i(s) = \sum_{c|i \in c} V_c(x_c) \tag{4.66}$$

where c is a clique of graph G, V_c is a potential function and Z is a normalizing constant. A clique is a subset of sites S where all sites are neighbors of each other.

We suppose an isotropic field so that:

$$U_i(s) = \sum_{j \in \mathcal{N}_i} V(s_i, s_j)$$

To reduce the sample space, constraints on the potential function V are introduced. A frequently used model is the Ising model. The potential functions are given by:

$$V(s_i, s_j) = \beta(1 - 2\delta(s_i, s_j)) \qquad \beta \neq 0 \tag{4.67}$$

where $\delta(0) = 1$ and $\delta(x) = 0$ for $x \neq 0$.

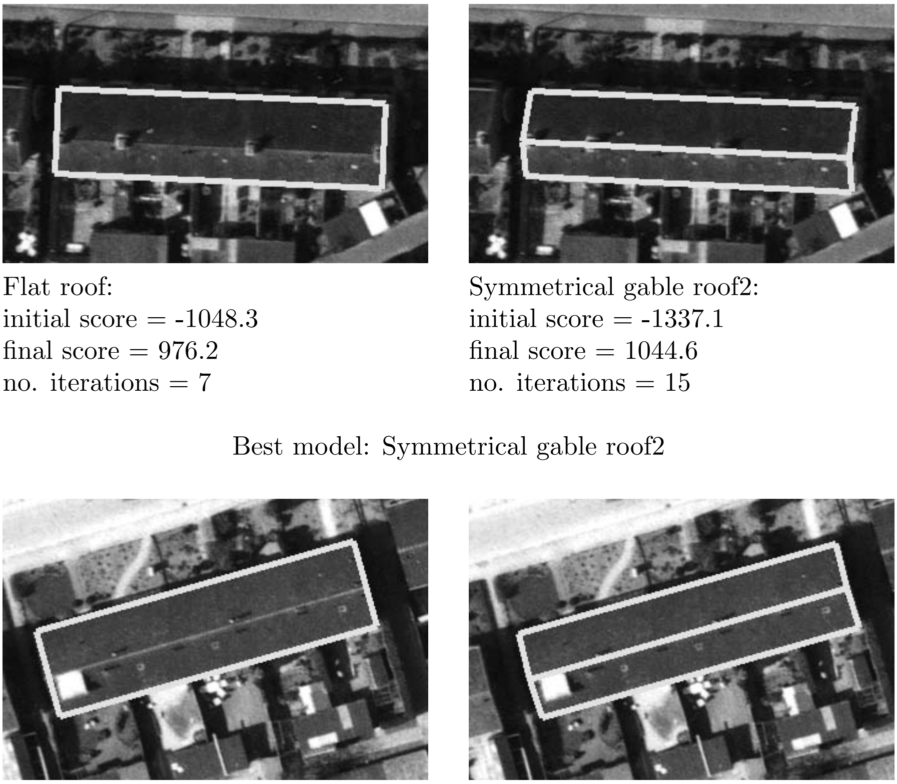
There are two possible values for V for a given s_i. These values are $\{-\beta, \beta\}$. In case of the four neighbor system the number of possible values for U is 5. These values are: $\{-4\beta, -2\beta, 0, 2\beta, 4\beta\}$. Thus, we only have to consider 5 states instead of 256^4 configurations.

In this way the sample space is greatly reduced, but still requires a lot of training samples in order to have sufficient statistical data for computing the probabilities from equation (4.64). Hence, we use the isolated pixel model that can be computed more easily.

The mutual information as defined in (4.61) can be used to compare models with the same complexity, but the mutual information between a model and images increases with the complexity of the model. Therefore, the mutual information between different building models and image data are not directly comparable and cannot be used as an evaluation function. The problem can be solved by implying the MDL principle. Using (4.37), the score of model contours is given by:

$$S_{contour} = I_{contour}(M; D) - L(M) \tag{4.68}$$

Figure 4.6 shows some experiments for selecting the best building model from the list of building hypotheses. Although only the mutual information of contours computed for matching the hypotheses against images is used, the correct models are still reliably found.



Flat roof:
initial score = -1048.3
final score = 976.2
no. iterations = 7

Symmetrical gable roof2:
initial score = -1337.1
final score = 1044.6
no. iterations = 15

Best model: Symmetrical gable roof2

Flat roof:
initial score = -501.1
final score = 862.5
no. iterations = 10

Symmetrical gable roof1:
initial score = -539.7
final score = 993.6
no. iterations = 6

Best model: Symmetrical gable roof1

Figure 4.6: Evaluation of building model contours

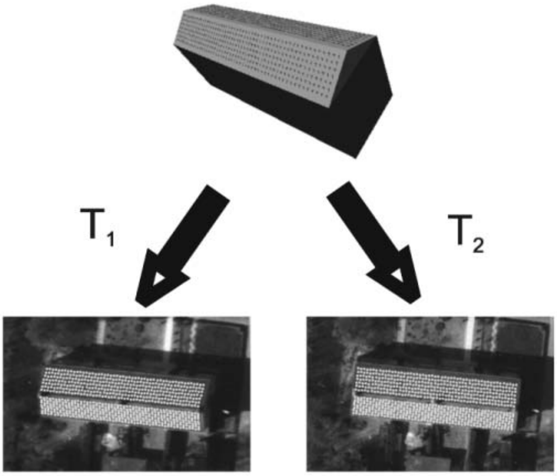


Figure 4.7: Grid used for computing the mutual information of the texture between two images. T_1 and T_2 are projection transformations of the 3D model into the two images

4.4.2 Mutual Information of Texture

If more images are available, the evaluation function can be further extended by including texture information. Given an object hypothesis and its pose in 3D, a point-to-point mapping can be defined between two images. If the object hypothesis is correct then the intensities at corresponding image pixels will be highly correlated.

If a and b are image intensity values of a pair of corresponding pixels in two images, then the mutual information of them based on (4.57) is:

$$I(a;b) = \log \frac{P(a,b)}{P(a)P(b)} \tag{4.69}$$

For computing the mutual information between the texture of two images, a grid is defined over the model (Figure 4.7). Then the mutual information is found by taking the sum of the mutual information of corresponding grid points from the two images over all grid points which are visible in both images:

$$I_{texture}(D_1;D_2|M) = \sum_{p \in grid} \log \frac{P(T_1(p),T_2(p))}{P(T_1(p))P(T_2(p))} \tag{4.70}$$

Building1:

Flat	$S_{contour} = 976.2$	$S_{texture} = 136.3$	$S_{final} = 1112.5$
Sym gable2	$S_{contour} = 1044.6$	$S_{texture} = 597.2$	$S_{final} = 1641.8$

Building2:

Flat	$S_{contour} = 862.5$	$S_{texture} = 568.4$	$S_{final} = 1430.9$
Sym gable1	$S_{contour} = 993.6$	$S_{texture} = 919.3$	$S_{final} = 1912.9$

Table 4.2: Evaluation of building models

where $T_1(p)$ and $T_2(p)$ are intensity values at the locations of the projections in the two images of the grid point p.

The joint distribution and the marginal distribution of image intensities can be computed by training. In the training process no assumption of the correlation between image intensities is made. Therefore, the metric is highly independent of intensity transformation.

By introducing this score into the MDL framework, in the same way as we did with the mutual information of the contour, the score of texture information will be given by:

$$S_{texture} = I_{texture}(M; D) - L(M) \tag{4.71}$$

Finally, the total score between a model and images is given by:

$$S_{total} = S_{contour} + S_{texture} \tag{4.72}$$

The evaluation of building models from Figure 4.6 also including texture information is presented in Table 4.2. The introduction of texture information further strengthens the discriminating power of the evaluation function. Both the mutual information of the contour $I_{contour}$ and the mutual information of the texture $I_{texture}$ are the largest in case of the correct building model.

4.4.3 Estimating Densities for Building Evaluation

A key factor in evaluating building models is the estimation of the probability distributions from which the corresponding mutual information $I_{contour}$ and $I_{texture}$ are calculated. We need to know the a priori probability of the gradients at random image points $P(grad_i)$, the conditional probability of the gradients at model points $P(grad_i|point_m)$ and the joint image intensity probability of two images $P(a, b)$ respectively.

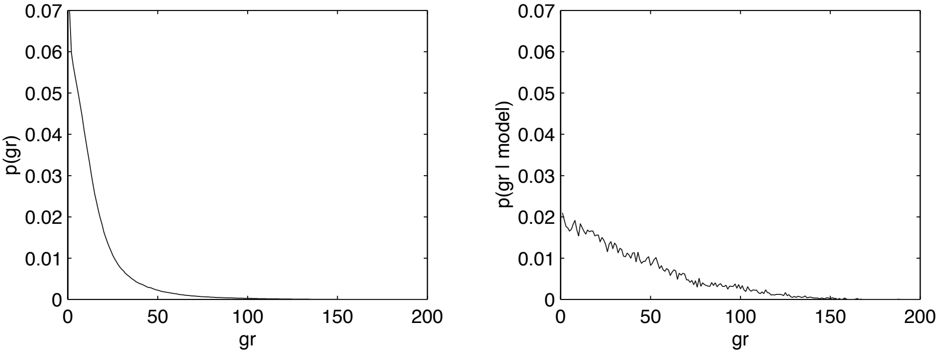


Figure 4.8: a) Gradient distribution $P(grad_i)$ b) Conditional probability density of the gradient $P(grad_i|point_m)$

The probability of the gradients at random image points can be obtained directly from the images, but training is necessary for estimating the conditional densities. The gradients are calculated in the regions of images where there is a building. The delineation of these regions in the images was described in chapter 2.2. Afterwards, these gradient values are used in the Parzen window estimation of the density. The obtained a priori probability $P(grad_i)$ is shown in Figure 4.8a.

The conditional probability density function of the gradients along the projected roof edges can be determined from training matches by analyzing the probabilities of gradients in these training matches. Some image lines corresponding to model lines are selected manually. Next, the gradient values along these lines are computed and used in the Parzen window estimation of the density. The obtained conditional probability density function $P(grad_i|point_m)$ is shown in Figure 4.8b.

The densities are estimated by a non-parametric method, Parzen window density estimate. For N sample points x_n, the density take the following form:

$$p(grad_i|point_m) = \frac{1}{N} \sum_{a \in x_n} K(grad_i - grad_a) \tag{4.73}$$

where the kernel function K is a Gaussian distribution.

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{4.74}$$

and σ is the standard deviation of the Gaussian.

Knowing these two distributions, the mutual information can be computed using (4.59) and this is shown in Figure 4.9.

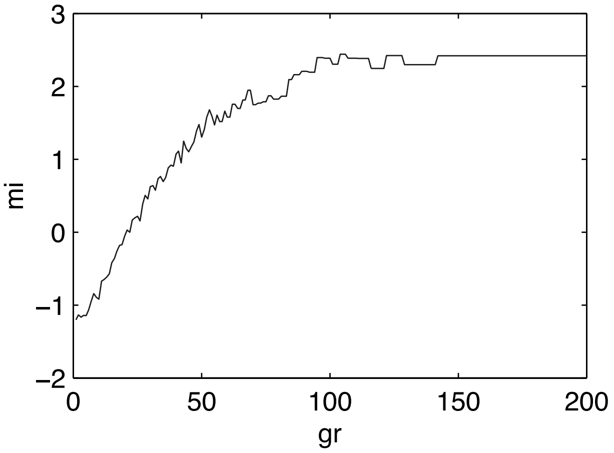


Figure 4.9: Mutual information $I(point_m; point_i)$

The technique is the same for estimating the joint probability distribution of the intensities $P(a, b)$ needed for the computation of the mutual information of texture (4.70). However, this time we have a two-dimensional case. N samples of corresponding points in the two images $x_n = \{(a_1, b_1), ..., (a_N, b_N)\}$ are chosen. These points can be chosen manually, but this would require huge effort. A more convenient method is to define a grid over some 3D building models already reconstructed and to project the points of this grid in both images in order to find corresponding points.

The density in an arbitrary point (a, b) will be given by:

$$p(a,b) = \frac{1}{N} \sum_{(a_i,b_i)\in x_n} g_\Sigma((a,b) - (a_i,b_i)) \tag{4.75}$$

with

$$g_\Sigma(x) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) \tag{4.76}$$

and the variance Σ is

$$\Sigma = \sigma^2 I \tag{4.77}$$

The joint probability of intensities $P(a, b)$ is shown in Figure 4.10.

Having computed the joint distribution, the marginal distributions of the intensity values in the two images can be found by:

$$p(a) = \sum_b p(a,b) \tag{4.78}$$

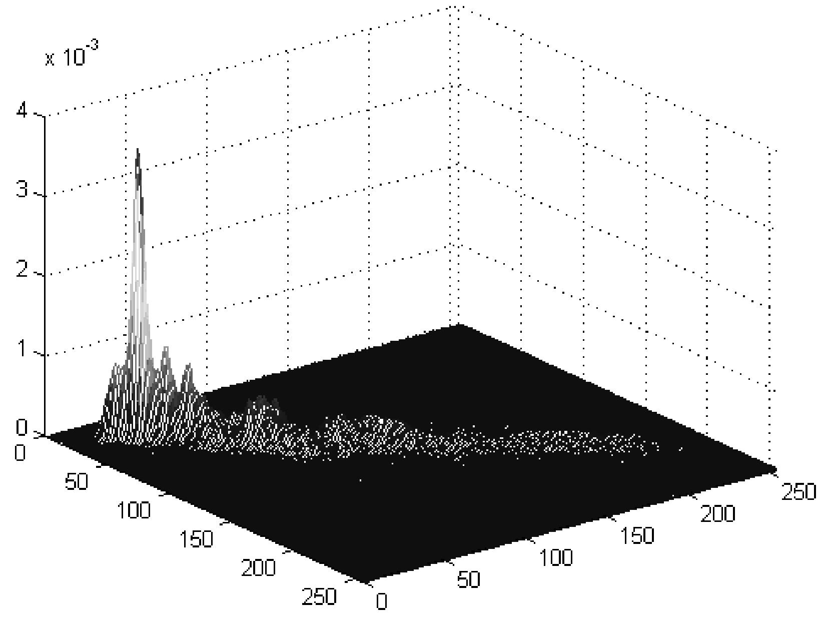


Figure 4.10: Joint probability of intensities $P(a, b)$

and

$$p(b) = \sum_{a} p(a, b) \tag{4.79}$$

Then, the mutual information of intensities can be computed using (4.69) and it is shown in Figure 4.11.

There are some factors that influence the result of the density estimation. Clearly, the choice of the variance of the Gaussian distribution σ affects the accuracy of the density estimate. Another factor is the interpolation method used for finding the intensity of a point. That is because the grid points projected into an image generally do not coincide with the image pixels, they can fall between four pixels in the image and interpolation is required.

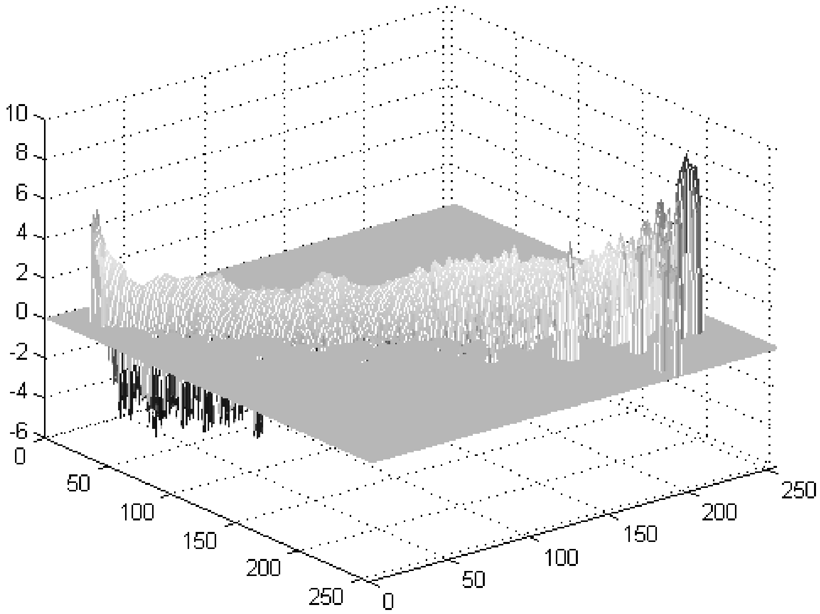


Figure 4.11: Mutual information of intensities $I(a;b)$

Estimating the Variance of the Kernel

Leave-one-out cross-validation is used for optimizing σ. The idea is to compute the evidence curve $v(\sigma)$ that approximately represents the likelihood of σ given the data. Then σ that maximizes $v(\sigma)$ is chosen. The steps required to compute $CV(\sigma)$ for a particular value of σ are the following:

1. The training data is split into two parts A and B, where A contains $N-1$ samples and B contains only a single sample in case of the leave-one-out validation method.

2. The Parzen window density estimate $p_a(x)$ is computed from A only.

$$p_a(x) = \frac{1}{N-1} \sum_{a \subseteq A} G(x - x_a) \tag{4.80}$$

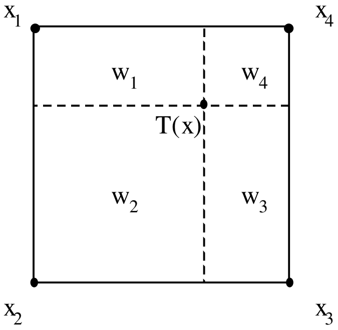


Figure 4.12: Pixel interpolation. x_1, x_2, x_3, x_4 are image pixels and $T(x)$ is the projection of the 3D point x

3. The log-probability of the sample which was held out $\log(p(B))$ is computed

$$\log p(B) = \log \frac{1}{N-1} \sum_{a \subseteq A} G(x_B - x_a) \tag{4.81}$$

4. Repeat from step 1 splitting the data differently. This is done for every possible choice of B.

5. The average of all log-probabilities obtained in step 3 is defined to be $v(\sigma)$.

$$CV(\sigma) = \frac{1}{N} \sum_{i=1}^{N} \log p(B_i) \tag{4.82}$$

These steps are repeated for different values of σ, then σ can be chosen such that $CV(\sigma)$ is maximum.

The downside of the Parzen windowing is that the scheme is very slow compared to traditional histogramming. However, since the density estimation for a pair of images can be done before the actually processing step, this expense matters little.

Interpolation

There are different interpolation methods, that can be used for determining the intensity of a projected grid point. Since interpolation affects the results of the density estimation, a number of interpolation methods are presented here.

The simplest method is the *nearest neighbor*. This method actually does not do an interpolation, instead it chooses the intensity of the closest image pixel.

A very popular method is *bilinear interpolation*. Bilinear interpolation estimation computes the intensity value of a point projected into image, $v^*(T(x))$, using the weighted average of the four nearest neighbors, where the weighting depends on the relative position of $T(x)$.

$$v^*(T(x)) = \sum w_i * v(x_i) \tag{4.83}$$

where x_i are the four neighbors of $T(x)$ in the image, the weights w_i are the surface areas indicated in the image, and $v(z)$ denote the intensity at pixel z (Figure 4.12).

Bilinear interpolation uses $v^*(T(x))$ in the density estimation (Figure 4.13 top).

Another interpolation method is *partial volume interpolation*. This method uses the same four nearest neighbors of $T(x)$, but instead of using one value $v^*(T(x))$ in the density estimation, it uses the four values $v(x_i)$. In case of histogramming, for each nearest neighbor x_i the histogram entry $v(x_i)$ is incremented by the weight w_i, where w_i is calculated as in the bilinear interpolation (Figure 4.13 bottom).

In this way, no new and therefore false, intensity values are formed. The process also keeps the changes to the histogram relatively small at each step. Partial volume interpolation updates multiple entries in the density function, creating dispersion, resulting in lower values in the density estimation. Subsequently, the mutual information decreases.

The effects of bilinear interpolation on mutual information are more varied. On one hand, if the image has relatively few gray values, the interpolation process creates new averaged gray values. These new values further disperse the histogram and increase joint entropy. On the other hand, if the image has many gray values, the interpolation process averages the diverse values into fewer values, reducing joint entropy and increasing mutual information. Real images have noise, which introduces many new values and creates the latter situation. Interpolation does not harm higher resolution images as much.

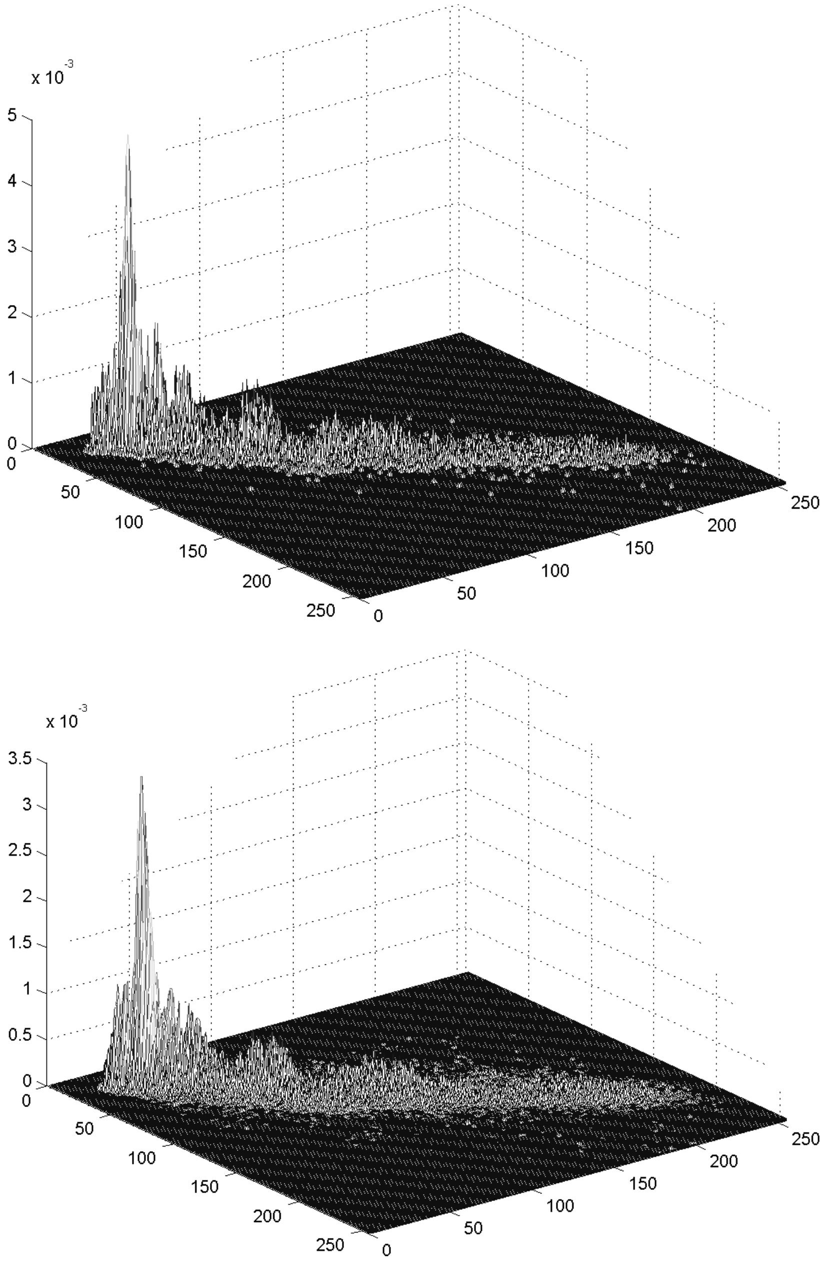


Figure 4.13: Interpolation methods. Top bilinear interpolation. Bottom partial volume interpolation.

Chapter 5

Experimental Results

This chapter contains a number of experiments designed to demonstrate the performance of the 3D building reconstruction system described in the previous chapters. First, the steps that have to be done to prepare the data for processing are presented. Experiments were carried out on two data sets. Experiments were first run on a data set with small variations in the terrain height. The second set of tests deals with the case of large variations in the terrain height. The chapter will conclude with an assessment of the performance of the approach.

5.1 Data Preprocessing

The data sets consist of pairs of aerial images and 2D GIS databases containing the ground plans of the buildings. There are a couple of problems that should be addressed before starting the 3D building model generation process.

First, the interior orientation parameters of the camera and also the exterior orientation parameters of the images should be known. The exterior orientation parameters should be given in the coordinate system of the map, to avoid an extra registration step. If the orientation parameters are not explicitly provided in the data set, then they can be computed using some control points (at least 4 for avoiding ambiguity) in case of the exterior orientation and using fiducial marks in case of the interior orientation.

Also, the variation of the terrain height has to be known a priori. This involves knowing the minimum and the maximum terrain height. These values are used to localize the buildings in the images. Depending on the variation of the terrain height one of two methods for localizing the buildings is chosen.

Finally, the probability density functions that are involved in the computation of the scores of the building models have to be determined. The a priori probability of the gradients at random image points $P(grad_i)$, the conditional probability of the gradients at model points $P(grad_i|point_m)$ and the joint image intensity probability of two images $P(a,b)$ can be found using the procedures described in detail in section 4.4.3.

5.2 Small Height Variations of the Terrain

The test data consists of high-resolution aerial images from Wijhe (The Netherlands). The scale of the images is 1:3000 and they are scanned at 600 dpi. Therefore, one pixel in image corresponds to about 12.7cm in object space. Two images with 60% overlap are used. The variation of the height is small (i.e less than 15 m), so the terrain can be considered as flat.

The first experiment was to generate building models for simple buildings composed of only one building primitive. In the current implementation 5 models are generated corresponding to a flat roof building primitive and two symmetrical gable roof primitives and two non-symmetrical gable roof primitives with two different orientations. Therefore we can reconstruct only flat roof buildings, gable roof buildings or buildings formed by combining these two building types. However the building library can be easily extended with other primitive building models.

The first step is the localization of buildings in the images. Since the height variations of the terrain for this data are small, the region of interest can be determined with the method described in section 2.2.2. The regions of interest in the two images found for the building are shown in Figure 5.1. This allow us to delineate the regions of interest of the wall primitives (Figure 5.2) and to label the extracted image features (Figure 5.3) as wall primitives and roof primitives. Next, the image corners, labelled as wall primitives from the two images are matched and 3D corners are generated. These corners are then used to determine the initial height of the building primitive required in the fitting procedure. First a flat roof primitive is fit to the image data and then its score is computed (Figure 5.4 top). Next, different gable roof primitives are considered. Since the length of the primitive is much larger than its width, only the gable roof primitives oriented along the larger edge of the building are generated (Figure 5.4 bottom). The gable roof primitives oriented along the small edge of the building are not considered at all. If the real roof is a non-symmetrical gable roof, then fitting a symmetrical gable roof produces either an invalid shape or a roof with very negative score. So, if the score of the fitting of a symmetrical gable roof is positive, then the case of a non-symmetrical gable does not have to be considered. This is the case in this experiment. Since the score of the fitting of a symmetrical gable roof primitive is high ($S_{total} = 2115.4$), the non-symmetrical gable roof primitive is not processed.

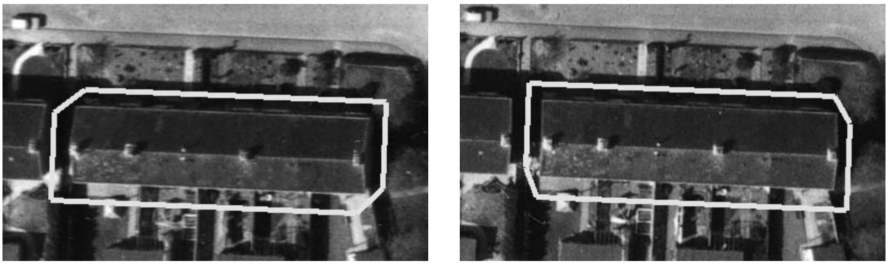


Figure 5.1: Localization of the building

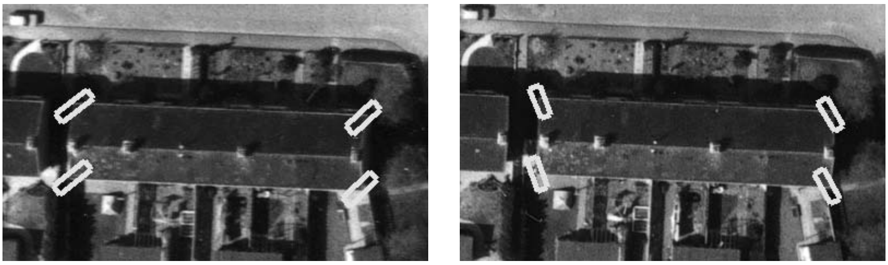


Figure 5.2: Localization of the roof corners

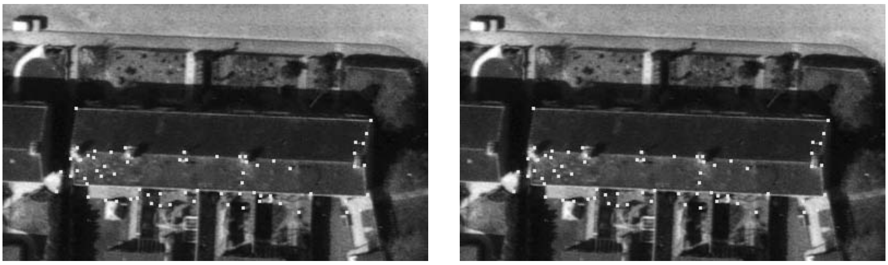
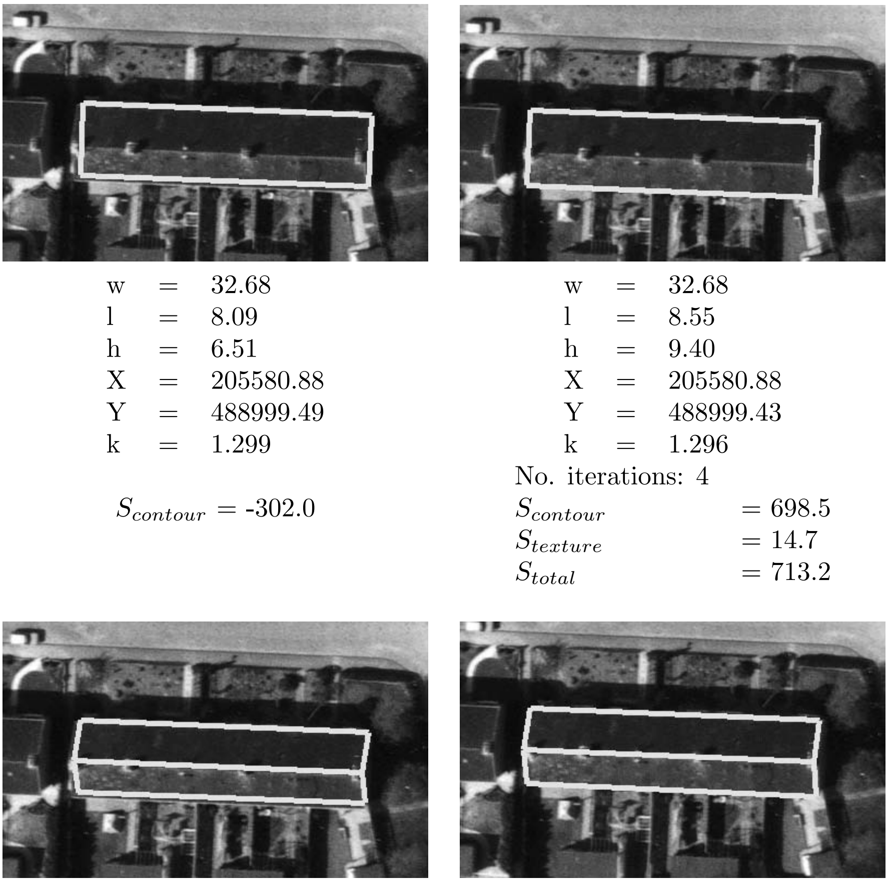


Figure 5.3: Extracted interest points

Note that during the fitting, only the mutual information of the contour $S_{contour}$ is computed. The mutual information of the texture $S_{texture}$ is computed only after the fitting stops. The fitting stops either when it converges, or otherwise after a predefined number of iterations.

Next, experiments with complex buildings are presented. First the partitioning of the building based on the ground plan is performed (Figure 5.5). The building can be divided into 4 simple partitions. These simple partitions can be combined and



w	=	32.68	w	=	32.68
l	=	8.09	l	=	8.55
h	=	6.51	h	=	9.40
X	=	205580.88	X	=	205580.88
Y	=	488999.49	Y	=	488999.43
k	=	1.299	k	=	1.296
			No. iterations: 4		
$S_{contour}$	=	-302.0	$S_{contour}$	=	698.5
			$S_{texture}$	=	14.7
			S_{total}	=	713.2

w	=	32.68	w	=	32.76
l	=	8.09	l	=	8.58
h	=	6.51	h	=	9.38
X	=	205580.88	X	=	205580.89
Y	=	488999.49	Y	=	488999.41
k	=	1.299	k	=	1.295
hr	=	5.3	hr	=	2.38
			No. iterations: 5		
$S_{contour}$	=	-600.7	$S_{contour}$	=	1636.0
			$S_{texture}$	=	4779.4
			S_{total}	=	2115.4

Best model: Gable roof primitive.

Figure 5.4: Generation of a simple building model. Left initial values of the parameters. Right values of the parameter after fitting. Top flat roof model. Bottom symmetrical gable roof model.

composed partitions are obtained. There are 4 composed partitions. These partitions lead to 5 possible partitioning schemes for the given building. In the next step for each partition building hypotheses are generated, corresponding to the building primitives defined in the building library. By evaluating the hypotheses generated for each partition, the best models are found and these are shown in Figure 5.6. The CSG trees describing every partitioning scheme are constructed. The leaves of these trees contain the best building models corresponding to each partition. In the final step, the CSG trees are fit to images. The geometric constraints between the partitions of a partitioning scheme are integrated into the fitting. The results of fitting the CSG trees corresponding to the five possible partitioning schemes are shown in Figure 5.7. The partitioning scheme with the highest score is selected. This is the first partitioning scheme composed of a flat roof primitive and a symmetrical gable roof primitive. The 3D model corresponding to the building is presented in Figure 5.8.

The problem with this approach is that a large number of partitions (8) and partitioning schemes (5) have to be treated. The number of partitions can be reduced by introducing partitioning rules. By applying the partitioning rule that divides the building into two parts by extending two existing collinear edges of the ground plan, only two partitions forming a single partition scheme are obtained. This partitioning scheme is exactly the best scheme obtained with the previous approach.

Another example of complex building is shown in Figure 5.9. The difficulty here is given by the fact that the small shed is completely shadowed by the main building. By applying the same partitioning rule as in the previous example, the building is divided in 4 partitions. It is clear that the very narrow partition along the top side of the ground plan ($w < 1$) cannot form a valid partition, therefore it has to be combined with another partition. In this way, only a single partitioning scheme is obtained. Figure 5.10 shows the building models with the highest scores corresponding to each of the 3 partitions forming the partition scheme. Figure 5.11 shows the reconstructed building model. The position of the small shed is stabilized by connecting it to the main building using a connection constraint.

More results on the reconstruction of complex building models are presented in Figure 5.12. The first three building models are reconstructed correctly, this is indicated by their high scores. The last building model seems to be wrong. The flat roof might be actually a gable roof. This uncertainty is reflected by the negative score of the texture information.

Figure 5.13 presents the results from a part of the scene.

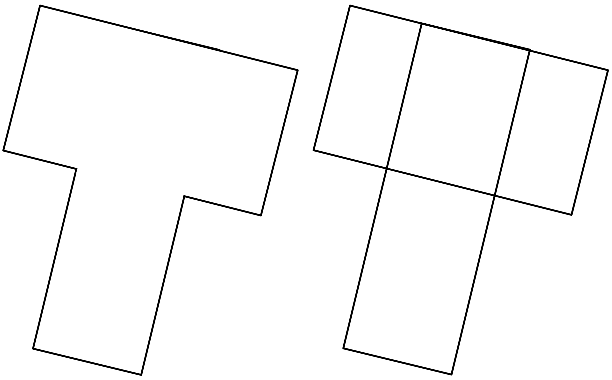
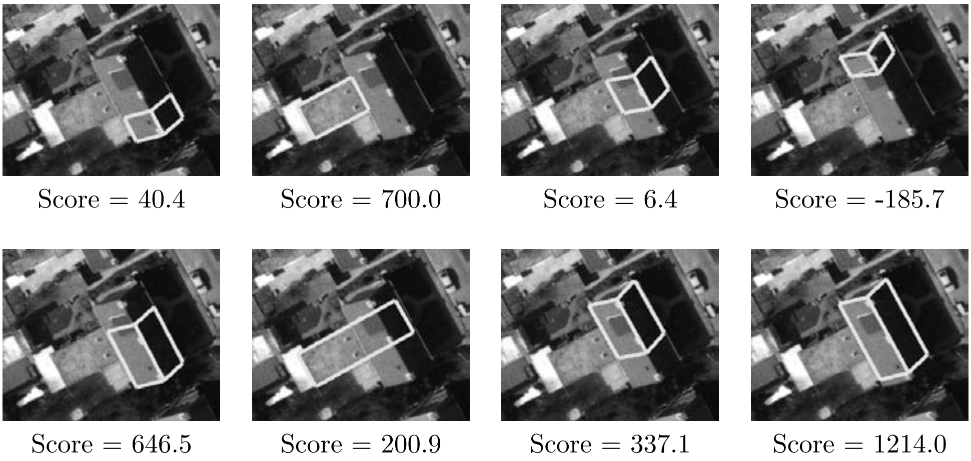


Figure 5.5: Ground plan of the building and its partition



Score = 40.4

Score = 700.0

Score = 6.4

Score = -185.7

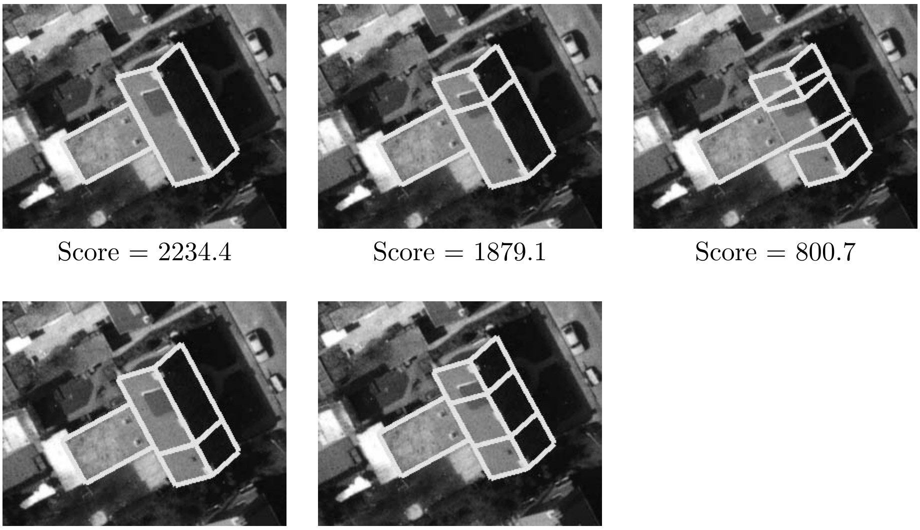
Score = 646.5

Score = 200.9

Score = 337.1

Score = 1214.0

Figure 5.6: Generation of building models for the partitions



Score = 2234.4

Score = 1879.1

Score = 800.7

Score = 1821.5

Score = 1325.2

Figure 5.7: Building models corresponding to the possible partitioning schemes

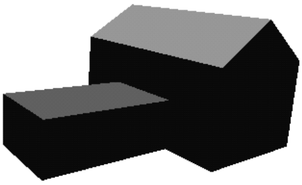


Figure 5.8: Vrml model of the best building model (Score = 2234.4)

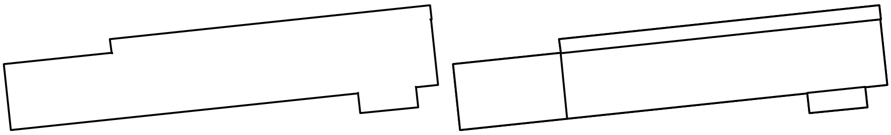
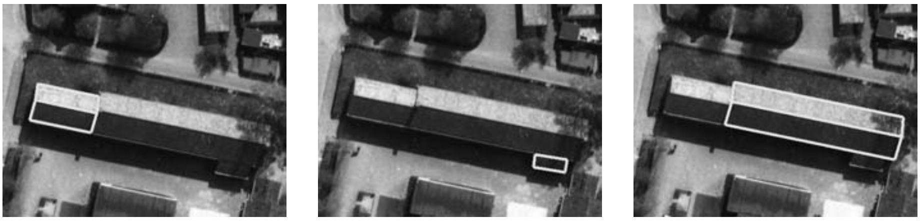
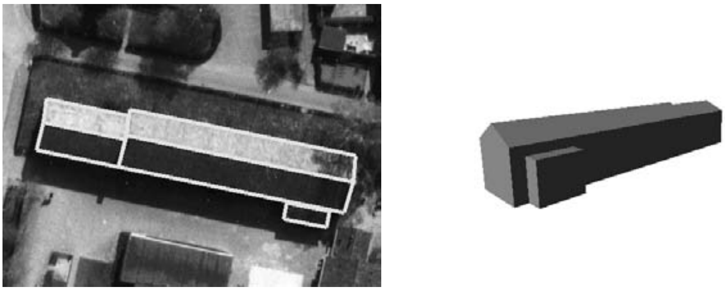


Figure 5.9: Ground plan of the building and its partitioning



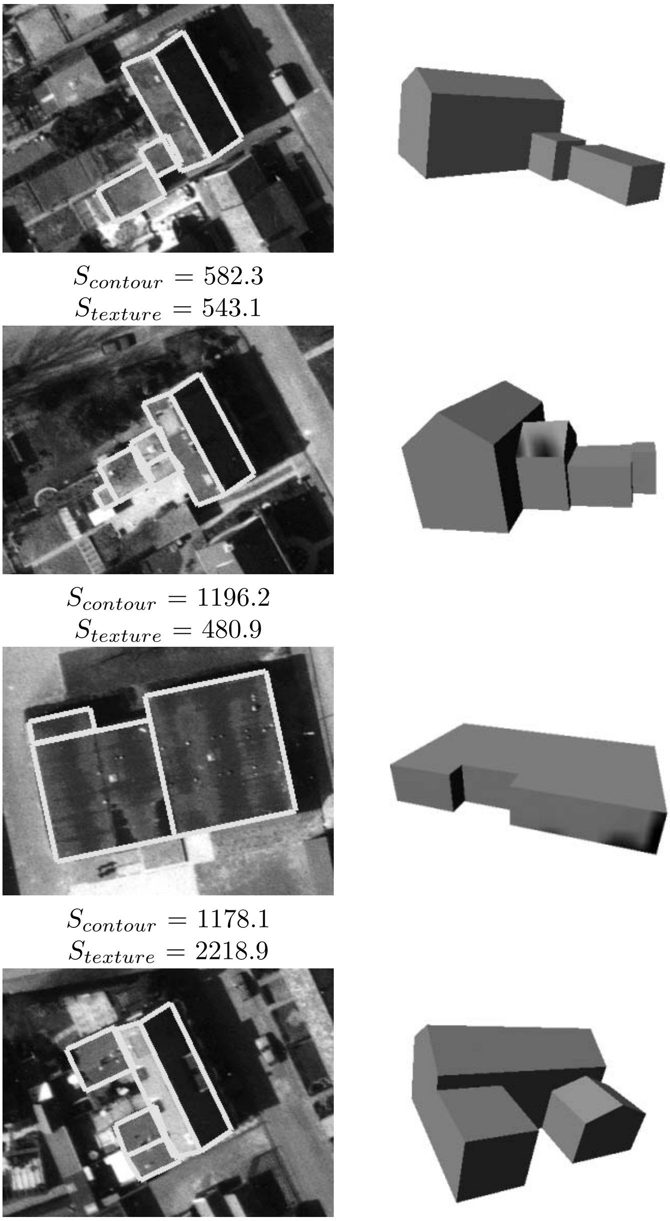
Score = 1456.0 Score = -41.9 Score = 3688.0

Figure 5.10: Generation of building models for the partitions



$S_{contour} = 3001.7$
$S_{texture} = 3001.2$
$S_{total} = 6002.9$

Figure 5.11: Building model of the best partitioning scheme



$S_{contour} = 582.3$
$S_{texture} = 543.1$

$S_{contour} = 1196.2$
$S_{texture} = 480.9$

$S_{contour} = 1178.1$
$S_{texture} = 2218.9$

$S_{contour} = 197.2$
$S_{texture} = $ -192.9

Figure 5.12: Examples of reconstructed models of complex buildings

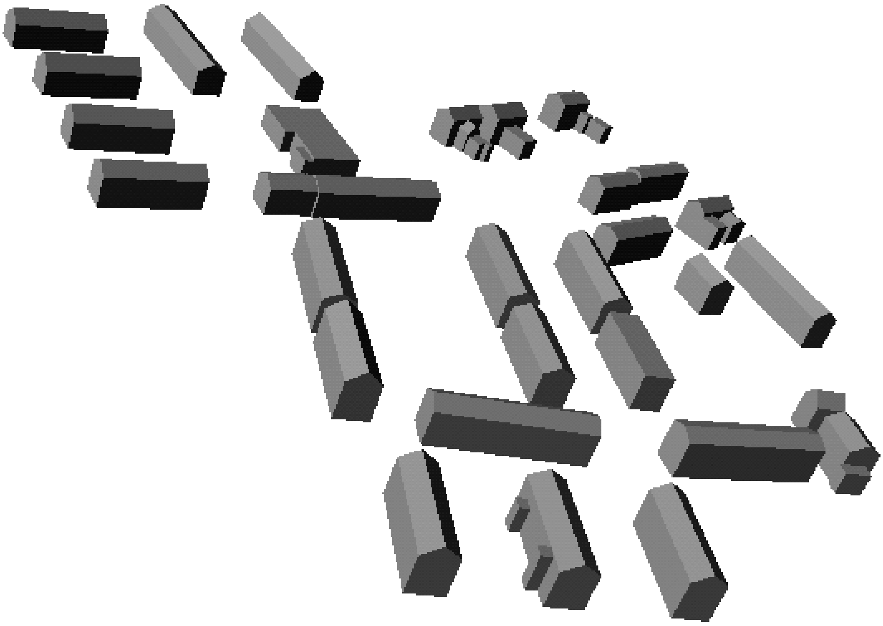


Figure 5.13: 3D model of a reconstructed scene

5.3 Large Height Variations of the Terrain

The test data consists of high-resolution aerial images of Stuttgart (Germany). The scale of the images is 1:13000 and they are scanned at 15 microns. This means that one pixel in image corresponds to about 20cm in object space. There are large height variations in the terrain, the minimum height is 220m and the maximum height is 260m. Therefore, the buildings cannot be localized using only map information and image information must also be considered.

Unfortunately, the quality of the images is very poor, the scanning of the photographs has introduced a lot of noise into the images. Since the photographs were scanned with a very high number of dots-per-inch, the resulting images contain a lot of grain (Figure 5.14). The grain affects our evaluation algorithm, especially the evaluation of the texture information. This evaluation of texture information is based on correlating image intensities from two images. However, with grain in both images, no correlation between the intensities of the two images can be established. Therefore, the score of the texture information is not computed, and

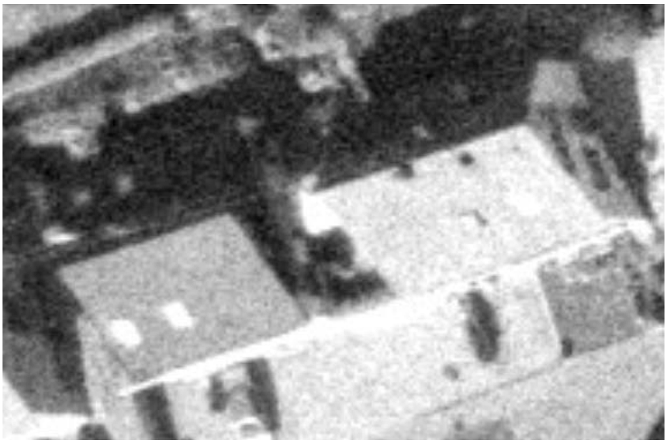


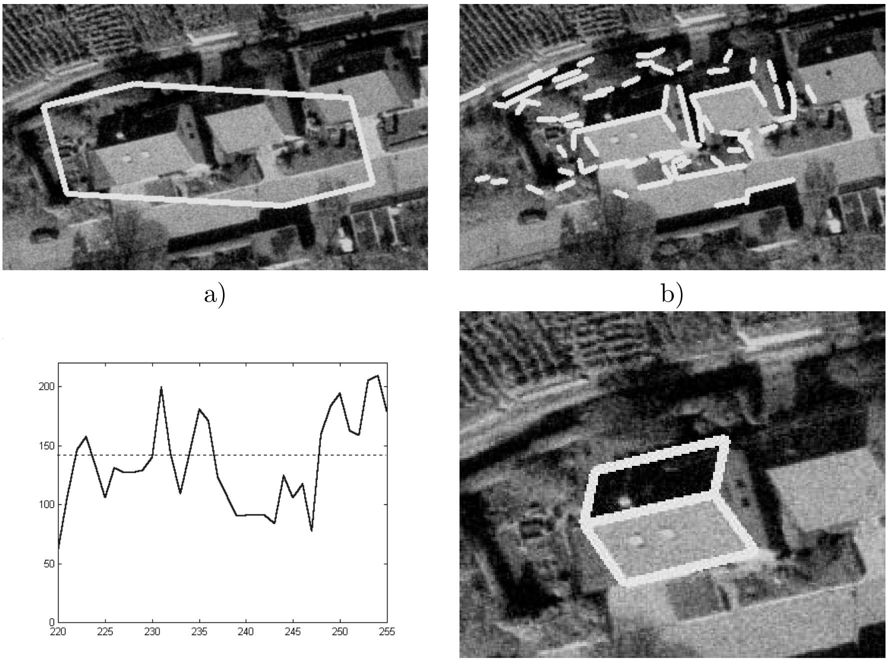
Figure 5.14: Digitization noise resulting in a grainy image

the evaluation of the building models is based only on the score of the building contours.

First the region of interest for a building is determined using map data and the minimum and maximum height values. This region is shown in Figure 5.15a. Afterwards, the possible locations of the building inside the ROI are verified by looking for image lines which could support each of the locations. The lines extracted from one of the images are shown in Figure 5.15b. The likelihood of the building location for different height values is given in Figure 5.15c. The most likely height values are verified by generating building hypotheses corresponding to the primitives defined in the building library and fitting them to the image data. The scores computed for matching the hypotheses against the images are used to select the best model. The table from Figure 5.15 shows the scores computed for different height values for the 5 building primitives. The best model obtained in this way, projected back into one of the images, is presented in Figure 5.15d.

More results for the localization and reconstruction of buildings composed of only one primitive are shown in Figure 5.16. Note that not all the possible hypotheses corresponding to all primitives from the library are generated. For instance, if a gable roof model oriented along the larger edge of the building is found, then the search stops and other gable roof hypotheses are not generated at all.

Next, complex buildings composed from more than one building primitive are considered. In the localization process the most likely height values are determined in the same way as for simple buildings. The partitioning of the building into building primitives based on the ground plan is performed. Then, for each resultant



a)

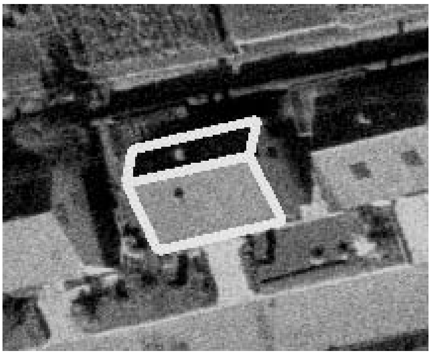
b)

c)

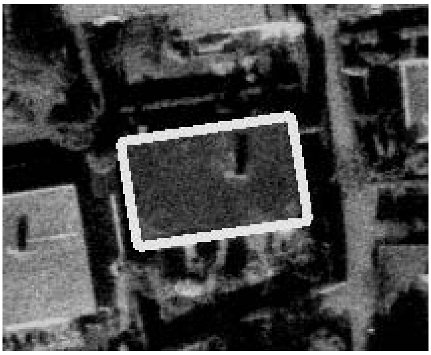
d)

height	flat	symgable1	gable1	symgable2	gable2
223	-249.0	-385.0	-397.4	-291.8	-203.5
231	-236.7	-282.1	-399.6	-268.7	-180.6
235	-288.1	-352.3	-398.7	-76.1	-197.8
250	-186.9	-313.0	-301.8	**155.3**	-
254	-214.8	-	-	58.6	-

Figure 5.15: Localization and reconstruction of a building

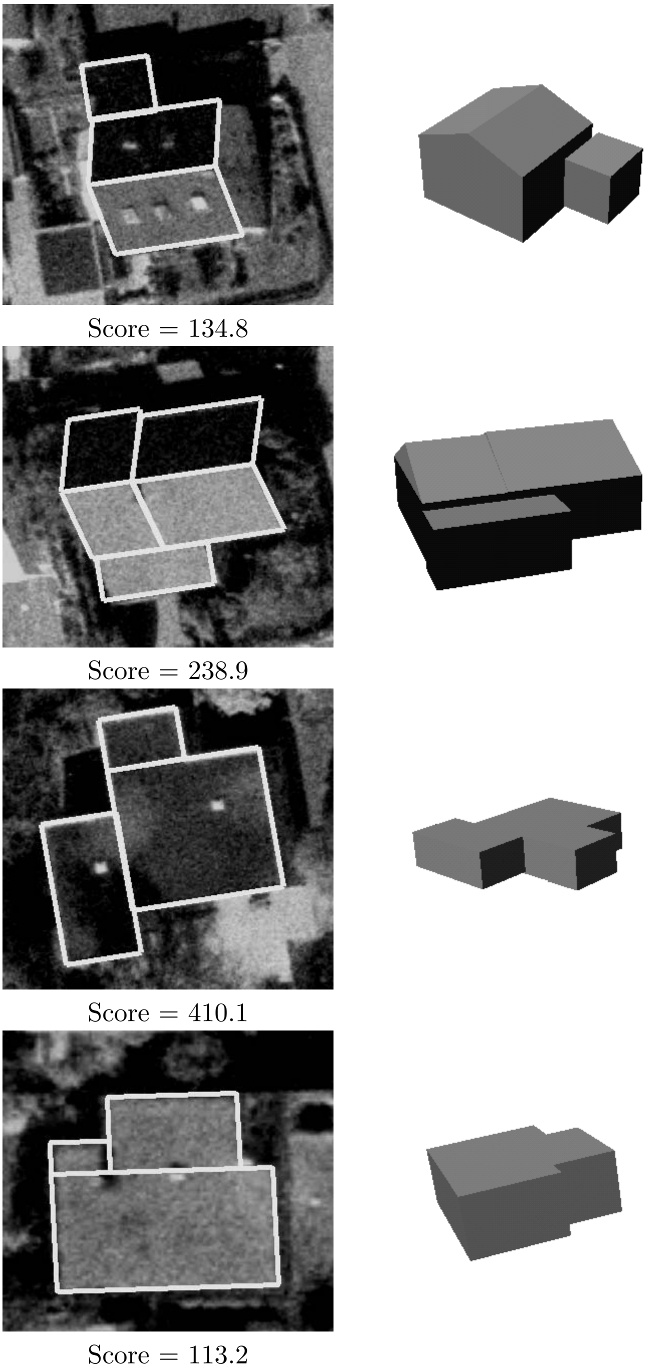


height	flat	symgable1	gable1	symgable2	gable2
220	-242.7	-329.7	-375.3	-318.2	-282.5
224	-230.2	-282.8	-345.7	-264.2	-174.5
250	-168.3	-	-	-258.0	**55.6**
255	-162.9	-	-	-286.1	-58.0



height	flat	symgable1	gable1	symgable2	gable2
222	**502.4**	223.7	-	360.2	-
227	431.1	-	-	187.4	-
253	-39.5	-	-	-74.5	- 44.5

Figure 5.16: Localization and reconstruction of buildings and associated scores for different building primitives and height values



Score = 134.8

Score = 238.9

Score = 410.1

Score = 113.2

Figure 5.17: Reconstructed models of complex buildings

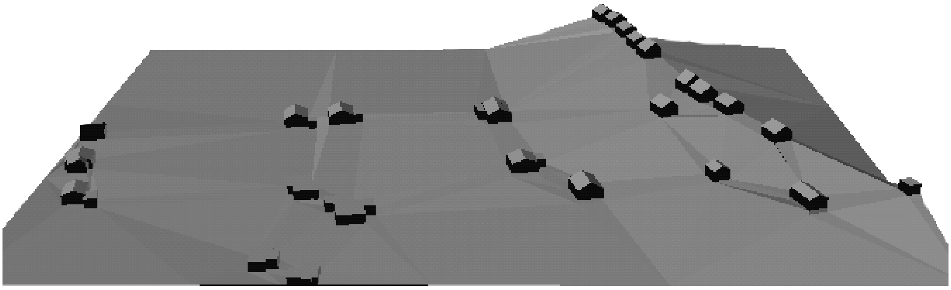


Figure 5.18: 3D model of a reconstructed scene

building partition, different hypotheses should be generated corresponding to every possible height value and every primitive from the building library. However, constructing and evaluating the whole search tree in this way would be rather time consuming. It is clear that the terrain under a given building can be considered flat. Therefore, it is enough to model the largest partition by the method described for simple buildings. The height of best building model corresponding to this partition can be used to generated building hypotheses for the rest of the partitions. The CSG tree describing the building hypothesis corresponding to a partitioning scheme is further refined by simultaneous fitting of the building models contained in the CSG tree. Finally, the CSG tree with the highest score is selected as building model. Figure 5.17 shows reconstructed models for a sample of complex buildings. Figure 5.18 presents the results from a part of the reconstructed scene.

5.4 Performance Assessment

The experiments carried out on the data sets presented in the previous sections were designed to address problems like the reconstruction rate and the accuracy of the reconstruction. No ground truth data is available for any of the data sets. Therefore, the generated building models can be verified only by visual inspection.

The evaluation function composed from two metrics $S_{contour}$ and $S_{texture}$ can be used to implement the "traffic-light" approach of classifying a reconstructed scene. One of the metrics, $S_{contour}$ can be used for verifying the resulting 3D model and the other metric $S_{texture}$ can be used for validation purposes. Therefore, depending on the two metrics a generated building model will be classified in the following way:

- green: both metrics are high (positive): the generated building model is

Class	No. buildings	No. correct models
green	30	29
yellow	7	6
red	6	0

Table 5.1: Classification of the reconstructed building models

certainly good.

- yellow: $S_{texture}$ is negative, $S_{contour}$ is positive: the generated building model may be good, but the system is not sure about the result.
- red: both metrics are negative: the building model is wrong or the building cannot be reconstructed by the system.

The scene from the first data set used for evaluation contains 43 buildings. The classification results obtained for these scene are summarized in Table 5.1. In each case the number of objects classified by the system in that specific category and the number of correctly reconstructed building models. There is one wrong building model classified as "green". The wrong building model is composed of a large main building and a small shed. The shed is reconstructed as a gable roof instead of the true flat roof, but the final score is not affected by this. Most of the "yellow" buildings are actually correct models, but their $S_{texture}$ score is negative due to unmodelled small roof structures such as small dormers and chimneys. The "red" class contains the buildings which cannot be reconstructed either because they are too small or because they lead to too many partitioning schemes. So, from 43 buildings 35 were reconstructed correctly and only 8 could not be reconstructed or the generated model is wrong.

Since no ground truth data is available, the verification of the accuracy of the reconstructed models is rather problematic. However, some accuracy measures can be derived from the variations of the parameters of the building models belonging to an area with the same design. The four buildings from the left top part of the Figure 5.13 form such an area. Within this area the buildings are assumed to be symmetrical gable roof buildings having the same width, length, orientation and gable height. The average parameters and the root mean square (RMS) obtained for these buildings are shown in Table 5.2. The deviations are less than 10 cm for the length, width and gable height, but the deviation for the height is quite large. This is due to the fact that the roof base edges are difficult to detect because of low contrast.

The computation time per building was in order of 5 seconds for simple buildings and in order of 1 minute for buildings with multiple partitions on a 1GHz Linux PC. This time does not include the time needed to read the images.

	width	length	height	gable height
average	8.49	26.48	7.71	10.87
RMS	0.04	0.03	0.38	0.08

Table 5.2: Average and RMS of the parameters of identical building models

The second test data contained 38 buildings. In this case, the texture information was not computed because of grain in the images. Therefore, a classification of the reconstructed building models in a traffic light manner could not be done. The results obtained in this case were not as good as in case of the first data set. Only 26 buildings were reconstructed correctly, 8 wrong building models were generated and 4 buildings were too complex. These results can be explained by reduced resolution of the images and also by the extra difficulty introduced by the large height variations in the terrain. The computation time per building also increased, since more hypotheses were generated. The computation time was in order of 20 seconds for simple buildings and in order of 2 minutes for complex buildings.

Chapter 6

Conclusions

A new approach to generate 3D building models from aerial images has been developed. The approach integrates the aerial images analysis with information from a GIS database and domain knowledge. The conclusions will summarize what has been presented, pointing out the strengths and the weaknesses of our approach. The recommendations for future work are discussed in the last section.

6.1 Summary

The goal of this project was to develop an automatic system for the 3D reconstruction of buildings from aerial images. In the past a lot of work has been done in the field of reconstructing buildings. There are a couple of aspects that make our approach different from the rest. First, our method requires only pairs of stereo images compared to other systems that work with multiple images which are not yet available for many countries. Our goal was to design a system that can work in dense urban areas where scene clutter and variety of building types complicate the process. To reduce the difficulty of the reconstruction, we used large-scale 2D GIS databases as additional information sources. GIS databases are widely available for most well-developed countries. Combination of image data and map data turned out to improve the reliability of the reconstruction. Generic knowledge about the shape of the buildings is also incorporated in the system. Since most buildings can be described as an aggregation of simple building types, the knowledge about the problem domain can be represented in a building library containing simple building models. Therefore, a building library was defined containing the most common building primitives, such as flat roof, and different types of gable roofs.

The building reconstruction process was formulated as a multi-level hypothesis generation and verification scheme and it was implemented as a search tree. The principal technical contribution of the work consisted of three aspects. First, a method that can localize the buildings in images was developed. In case of a flat terrain model, the region of interest of the buildings can be determined exactly. Otherwise a set of possible height values are returned and these values are verified by a higher level process.

Secondly, generation of building hypotheses corresponding to the primitives defined in the building library was carried out. This implies stereo matching of image features (corners, lines) that correspond to map primitives. The building hypotheses are refined by use of a fitting algorithm which fits the edges of the projected wire-frame of the model to the image gradients. The same fitting algorithm was also used for fitting complete building models described as CSG trees in the final hypothesis verification step. This time constraints describing the geometric relationships between building primitive, are incorporated in the fitting algorithm.

An original contribution was the definition of a metric for evaluating the generated hypotheses in order to select the one that best described the image. The metric is based on the formulation of the mutual information between the building model and the images. Methods for the estimation of the mutual information from training samples were analyzed. Our metric combines two measures. One of them counts for evidence along the model contour in the images. Therefore, the mutual information of the image gradients along the model contours was learnt. The second measures image intensity similarity over two images. Therefore, the mutual information of the intensities at corresponding pixels from two images was trained.

These image/model comparison measures have been rigorously derived from information theory. The technique used for comparing 3D object models to images does not require a priori information about the surface properties of the object and is robust with respect to variations in illumination. Also, no assumption about the shape of the objects are made. As result the method is quite general and may be used in a wide variety of applications.

We have produced an approach that is able to meet most of the requirements described in the first chapter, and an implementation of this approach that provides good results. More importantly, it demonstrates the validity of some observations about the problem domain and shows some promising areas for further research.

6.2 Problems

The goal of our research was to design an entirely automated system. Given the extreme complexity of some images, often rather challenging even for expert human users, this goal may not be fully achievable. The reconstruction becomes difficult due to some related factors, such as high building density, complex building and surface shapes, little space between buildings. It is believed that the reconstruction can be automated given a large enough set of images. It is generally true that if sufficiently large number of appropriate image viewpoints are not available, any reconstruction problem can be difficult or impossible. Thus, expectations of a completely automated system must be tempered. However, we have tried to come as close as possible to our goal.

There are several problems, or deficiencies, that still exist in this system. Most of these are minor in the sense that their solution would be necessary for a production system, but would not add much in terms of theoretical advancement.

First, the partitioning of the buildings is performed using map data only. On one hand, this might lead to a large number of partitioning schemes in case of complex buildings. Many of these partitioning schemes are unlikely to occur in reality. Some partitioning rules were learnt. Two of these rules are actively used in the system. These rules were established by looking only at the map lines. More sophisticated rules could be probably be derived if the whole building structure is taken into account. On the other hand, the real partitioning of the building might not be found by using only map data. Therefore, image information should be considered. Fortunately, there are few buildings of this kind.

Other problems appear at the fitting of small primitives. Since, the small primitives do not have well defined edges in the images, the fitting algorithm may become instable. An attempt was made to stabilize the solution by increasing the weights of the initial values of parameters of the primitives, except the height value. The results improved but there were still situations when the fitting lead to inconsistent solutions.

The number of primitives defined in the building library affects the performance of the system. The system can reconstruct buildings that can be described as combination of the building primitives from the library. It would be useful to have a building library with many primitives. However, this would lead to a combinatorial increase of the processing time, since the number of primitives that have to be checked at the hypotheses generation step would grow. The number of primitives that have to be checked can be reduced by introducing some simple rules, for example only generate gable roofs with orientation along the larger edge if the larger edge is twice as large as the other edge.

The metric that measures image intensity similarity over two images can also raise problems. If the intensities of the images are very different, the mutual information of the intensities at corresponding pixels from two images will not carry too much information. That is because the joint probability density of the intensities from the two images for a wide range of values are similar.

6.3 Future Work

The observation of the regularities of the building structure could enhance the building reconstruction. This can be done by learning from image interpretation. In a traditional approach the learning is performed before the processing task by analyzing a training set of buildings. This method requires a large number of buildings to be analyzed. It is still possible that some not so frequent building structure cannot be learnt, since it is often impractical to obtain examples of building structures that are both correct and representative of all the situations. In this case, the learning would not produce reliable results.

This could be achieved by use of reinforcement learning. Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. The learning is not performed a priori, but during the processing and the learning algorithm must be able to learn from its own experience. For example, consider teaching a building structure: we cannot tell the system what to do, but we can give a reward or punishment if the generated model is respectively good or wrong.

Clearly, the generated 3D building models should be integrated into a 3D GIS database. 3D GIS databases are of great importance for many activities such as guided navigation, mobile communications, urban planning, virtual reality for tourism or architecture. The 3D models generated by our system, described as CSG trees with the leaf nodes containing the primitive building models that compose the building, can be easily converted into a representation that can be imported to the 3D GIS systems currently available on the market. The most known systems are the 3D Analysist of Arcview from ESRI Inc. and Imagine VirtualGIS from ERDAS. By integrating our models into a 3D GIS system, the user can make use of the capabilities such as analysis of the data, visualization and fly-through, that are already implemented in these systems. Also, the data already available in a 3D GIS database can provide useful information for the reconstruction of other buildings. This information can be treated as contextual information of the neighborhood of a building. Currently, our system processes each building separately. However, the neighborhood relation of the buildings should be taken into account. For instance, it is very unlikely to have a large jump in height between two close buildings. So, the height of the neighboring building or buildings, if they were already reconstructed could be considered as the initial value for the current building.

Laser scanner data is a valuable data source for 3D reconstruction of buildings. Recent advancements in airborne laser scanning (LIDAR) made possible the acquisition of dense point clouds. Hence, it is expected that future research on building reconstruction will be directed towards fusion of aerial images, laser scanning data and maps. Vosselman pointed out the advantages of such an approach [Vosselman, 2002]. These data sources have a complementary nature. Laser scanning data is suitable for the detection of planar surfaces, but the accuracy is low. On the contrary, edges can be delineated accurately in aerial images, but the extraction of the surface shape is often uncertain due to the lack of texture. Hence, the reconstruction could benefit from the combination of different data sources.

Bibliography

[Akaike, 1974] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

[Axelsson, 1998] Axelsson, P. (1998). Integrated sensors for improved 3D interpretation. In *Proceedings of IAPRS, ISPRS Commission III, Germany*, volume 32, Stuttgart, Germany.

[Ayache and Faugeras, 1986] Ayache, N. and Faugeras, O. D. (1986). HYPER: A new approach for the recognition and positioning of 2D objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54.

[Ayache and Faverjon, 1987] Ayache, N. and Faverjon, B. (1987). Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2).

[Baillard and Zisserman, 2000] Baillard, C. and Zisserman, A. (2000). A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, part b3, pages 56–62.

[Baltsavias et al., 2001] Baltsavias, E. P., Gruen, A., and Gool, L. V., editors (2001). *Automatic extraction of man-made objects from aerial and space images (III), Proceedings of Ascona Workshop 2001.* A.A. Balkema.

[Beveridge, 1993] Beveridge, J. R. (1993). *Local search algorithms for geometric object recognition: optimal correspondence and pose.* PhD thesis, University of Massachusetts.

[Beveridge et al., 1989] Beveridge, J. R., R.Weiss, and Riseman, E. M. (1989). Optimisation of 2D model matching. In *Proceedings of ARPA Image Understanding Workshop*, pages 815–830.

[Bignone et al., 1996] Bignone, F., O.Henricsson, P.Fua, and M.Stricker (1996). Automatic extraction of generic house roofs from high resolution aerial imagery.

In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 85–96.

[Brenner, 2000] Brenner, C. (2000). Towards fully automatic generation of city models. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, part b3, pages 85–92.

[Bubna and Stewart, 2000] Bubna, K. and Stewart, C. V. (2000). Model selection techniques and merging rules for range data segmentation algorithms. *Computer Vision and Image Understanding*, 39(1):1–38.

[Burns et al., 1986] Burns, J. B., Hanson, A. R., and Riseman, E. M. (1986). Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4).

[Collignon et al., 1995] Collignon, A., Vandermuelen, D., Suetens, P., and Marchal, G. (1995). 3D multi-modality medical image registration using feature space clustering. *Computer Vision, Virtual Reality and Robotics in Medicine*, pages 195–204.

[Collins et al., 1995] Collins, R. T., Jaynes, C. O., Cheng, Y.-Q., Wang, X. G., Stolle, F. R., Schultz, H., Hanson, A. R., and Riseman, E. M. (1995). The UMass Ascender system for 3D site model construction.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA.

[Deriche and Giraudon, 1993] Deriche, R. and Giraudon, G. (1993). A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101–124.

[Duda et al., 2001] Duda, R., Hart, P., and Stork., D. (2001). *Pattern Classification.* John Wiley & Sons, New York, second edition.

[Efron and Tibshirani, 1993] Efron, B. and Tibshirani, R. (1993). *An Introduction to Bootstrap.* Chapman and Hall, London, UK.

[Faugeras, 1995] Faugeras, O. (1995). Stratification of 3D vision: Projective, affine and metric representations. *Journal of Optical Soc. of America*, 12(3).

[Fischer et al., 1997] Fischer, A., Kolbe, T. H., and Lang, F. (1997). Integration of 2D and 3D reasoning for building reconstruction using a generic hierarchical model. In *Workshop Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, pages 159–180.

[Fischer et al., 1998] Fischer, A., Kolbe, T. H., Lang, F., Cremers, A. B., Foerstner, W., Plumer, L., and Steinhage, V. (1998). Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D. *Computer Vision and Image Understanding*, 72(2).

[Foerstner, 1994] Foerstner, W. (1994). A framework for low-level feature extraction. In *Proceedings of the European Conference on Computer Vision*, volume II, pages 383–394.

[Fradkin et al., 1999] Fradkin, M., Roux, M., and Maitre, H. (1999). Building detection from multiple views. In *Proceedings of ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume 32, Part 3-2W5, pages 81–86.

[Fua, 1997] Fua, P. (1997). *RADIUS: Image Understanding for Intelligence Imagery*, chapter Model-Based Optimization: An Approach to Fast, Accurate, and Consistent Site Modeling from Imagery. Morgan Kaufmann.

[Fua and Leclerc, 1990] Fua, P. and Leclerc, Y. (1990). Model driven edge detection. *Machine Vision and Applications*, 3:45–56.

[Girard et al., 1998] Girard, S., Guerin, P., Maitre, H., and Roux, M. (1998). Building detection from high resolution colour images. In *Proceedings of SPIE*.

[Grimson, 1990] Grimson, W. E. L. (1990). *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press.

[Haala, 1994] Haala, N. (1994). Detection of buildings by fusion of range and image data. In *Proceedings of IAPRS, ISPRS Commission III*, pages 341–346, Munich, Germany.

[Haala and Anders, 1996] Haala, N. and Anders, K. (1996). Fusion of 2D GIS and image data for 3D building reconstruction. In *International Archives of Photogrammetry and Remote Sensing*, volume 31, pages 289–290.

[Haala and Anders, 1997] Haala, N. and Anders, K. (1997). Acquisition of 3D urban models by analysis of aerial images, digital surface models and existing 2D building information. In *SPIE Conference on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision III*, pages 212–222.

[Hamming, 1980] Hamming, R. (1980). *Coding and Information Theory.* Prentice Hall.

[Hanson et al., 2001] Hanson, A., Marengoni, M., Schultz, H., Stolle, F., Riseman, E., and Jaynes, C. (2001). Ascender II: a framework for reconstruction of scenes from aerial images. In *Workshop Ascona 2001: Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), Ascona, Switzerland.*

[Hartley and Zisserman, 2001] Hartley, R. and Zisserman, A. (2001). *Multiple View Geometry in Computer Vision.* Cambridge University Press.

[Henricsson, 1996] Henricsson, O. (1996). *Analysis of image structures using color attributes and similarity relations.* PhD thesis, Swiss Federal Institute of Technology.

[Huertas et al., 1993] Huertas, A., Lin, C., and Nevatia, R. (1993). Detection of buildings from monocular views of aerial scenes using perceptual grouping and shadows. In *Proceedings of ARPA Image Understanding Workshop*, pages 253–260.

[Huertas and Nevatia, 1988] Huertas, A. and Nevatia, R. (1988). Detecting buildings in aerial images. *Computer Graphics and Image Processing*, 41:131–153.

[Huttenlocher et al., 1991] Huttenlocher, D., Kedem, K., Sharir, K., and Sharir, M. (1991). The upper envelope of voronoi surfaces and its applications. In *Proceedings of the Seventh ACM Symposium on Computational Geometry*, pages 194–293.

[Irvin and McKeown, 1989] Irvin, R. and McKeown, D. (1989). Methods for exploiting the relationships between buildings and their shadows in aerial imagery. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6).

[Jaynes et al., 1997] Jaynes, C., Marengoni, M., Hanson, A., Riseman, E., and Schultz, H. (1997). Knowledge-directed reconstruction from multiple aerial images. In *Proceedings of ARPA Image Understanding Workshop*, volume 2, pages 971–976.

[Jaynes et al., 1994] Jaynes, C., Stolle, F., and Collins, R. (1994). Task driven perceptual organization for extraction of rooftop polygons. In *Proceedings of ARPA Image Understanding Workshop.*

[Jaynes et al., 1996] Jaynes, C., Stolle, F., Schultz, H., Collins, R., Hanson, A., and Riseman, E. (1996). Three dimensional grouping and information fusion for site modeling from aerial images. In *Proceedings of ARPA Image Understanding Workshop*, volume 1, pages 479–490.

[Jibrini et al., 2000] Jibrini, H., Paparoditis, N., Deseilligny, M. P., and Maitre, H. (2000). Automatic building reconstruction from very high resolution aerial stereopairs using cadastral ground plans. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, part b3.

[Kanatani, 2000] Kanatani, K. (2000). *Data Segmentation and Model Selection for Computer Vision*, chapter Model selection criteria for geometric inference, pages 91–115. Springer-Verlag.

[Kass et al., 1987] Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

[Lang et al., 1995] Lang, F., Loecherbach, T., and Schickle, W. (1995). A one-eye stereo system for semi-automatic 3D-building extraction. *Geomatics Info Magazine.*

[Li, 2001] Li, S. (2001). *Stochastic Processes: Modeling and Simulation*, volume 20 of Handbook of Statistics, chapter Modeling Image Analysis Problems Using Markov Random Fields. Elsevier Science.

[Lowe, 1991] Lowe, D. (1991). Fitting parameterized 3D models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450.

[Lowe, 1987] Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(2):355–395.

[Maitre et al., 1995] Maitre, H., Bloch, I., Moissinac, H., and Gouinaud, C. (1995). Cooperative use of aerial images and maps for the interpretation of urban scenes. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images, Ascona.*

[McGlone and Shufelt, 1994] McGlone, J. and Shufelt, J. (1994). Projective and object space geometry for monocular building extraction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 46–53.

[Mohan and Nevatia, 1989a] Mohan, R. and Nevatia, R. (1989a). Segmentation and description based of scenes based on perceptual organization. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition.*

[Mohan and Nevatia, 1989b] Mohan, R. and Nevatia, R. (1989b). Using perceptual organization to extract 3D structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1121 –1139.

[Moons et al., 1998] Moons, T., Frere, D., Vandekerckhove, J., and Gool, L. (1998). Automatic modeling and 3D reconstruction of urban house roofs from high resolution aerial imagery. In *Proceedings of the European Conference on Computer Vision.*

[Noronha and Nevatia, 2001] Noronha, S. and Nevatia, R. (2001). Detection and modeling of buildings from multiple aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):501–518.

[Pasko and Gruber, 1996] Pasko, M. and Gruber, M. (1996). Fusion of 2D GIS data and aerial images for 3D building reconstruction. In *International Archives of Photogrammetry and Remote Sensing.*

[Pope and Lowe, 1994] Pope, A. R. and Lowe, D. G. (1994). Modeling positional uncertainty in object recognition. Technical Report TR-94-32, University of British Columbia.

[Press et al., 1992] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, second edition.

[Rissanen, 1978] Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

[Roux and McKeown, 1994] Roux, M. and McKeown, D. M. (1994). Feature matching for building extraction from multiple views. In *Proceedings of ARPA Image Understanding Workshop*, pages 331–339.

[Schmid and Zisserman, 1997] Schmid, C. and Zisserman, A. (1997). Automatic line matching across views. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition.*

[Schwarz, 1978] Schwarz, G. (1978). Estimating dimension of a model. *Annual of Statistics*, 6:461–464.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423 and 623–656.

[Shufelt and McKeown, 1993] Shufelt, J. and McKeown, D. (1993). Fusion of monocular cues to detect man-made structures in aerial imagery. *CVGIP: Image Understanding*, 57(3):307–330.

[Smith and Brady, 1997] Smith, S. M. and Brady, J. M. (1997). SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78.

[Steger, 2000] Steger, C. (2000). Subpixel-precise extraction of lines and edges. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIII, part b3, pages 141–156.

[Sullivan, 1992] Sullivan, G. D. (1992). Visual interpretation of known objects in constrained scenes. *Royal Society discussion meeting on Natural and artificial low level seeing systems.*

[Suveg and Vosselman, 2000] Suveg, I. and Vosselman, G. (2000). 3D reconstruction of building models. In *Proceedings of the XIXth congress of ISPRS*, volume XXXIII, part B2, pages 538–545.

[Suveg and Vosselman, 2001] Suveg, I. and Vosselman, G. (2001). 3D building reconstruction by map based generation and evaluation of hypotheses. In *Proceedings of the British Machine Vision Conference*, pages 643–652.

[Suveg and Vosselman, 2002a] Suveg, I. and Vosselman, G. (2002a). Automatic 3D reconstruction of buildings from aerial images. In *SPIE Photonics West, Electronic Imaging*, volume 4661, pages 59–69.

[Suveg and Vosselman, 2002b] Suveg, I. and Vosselman, G. (2002b). Mutual information based evaluation of object models. In *Proceedings of the International Conference on Pattern Recognition*, volume III, pages 557–560.

[Tangelder et al., 1999] Tangelder, J., P.Ermes, Vosselman, G., and van den Heuvel, F. (1999). Measurements of curved objects using gradient based fitting and CSG models. In *Proceedings of ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume 32, Issue 5-W11, pages 23–30.

[Torr, 1999] Torr, P. (1999). Model selection for a two view geometry - a review. In *Shape, Contour and Grouping in Computer Vision*, pages 277–301.

[Trucco and Verri, 1998] Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3D Computer Vision.* Prentice-Hall, Upper Saddle River, NJ.

[Tsai, 1987] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy vision metrology using off-the shelf TV cameras and lenses. *IEEE Transactions on Robotics and Automation*, 3(4):323–344.

[Viola and Wells, 1997] Viola, P. A. and Wells, W. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154.

[Vosselman, 1992] Vosselman, G. (1992). *Relational Matching.* Lecture Notes in Computer Science 628, Springer-Verlag.

[Vosselman, 2002] Vosselman, G. (2002). Fusion of laser scanning data, maps and aerial photographs for building reconstruction. In *Proceedings of International Geoscience and Remote Sensing Symposium*, volume I, pages 85–88, Toronto,, Canada.

[Vosselman and Suveg, 2001] Vosselman, G. and Suveg, I. (2001). Map based building reconstruction from laser data and images. In *Proceedings of Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*, pages 231–242, Ascona, Switzerland.

[Vosselman and Veldhuis, 1999] Vosselman, G. and Veldhuis, H. (1999). Mapping by dragging and fitting of wire-frame models. *Photogrammetric Engineering and Remote Sensing*, 65(7):769–776.

[Webb, 1999] Webb, A. (1999). *Statistical Pattern Recognition.* Arnold Publishers, London.

[Wells, 1993] Wells, W. M. (1993). *Statistical object recognition.* PhD thesis, Massachusetts Institute of Technology.

[Zhang, 1994] Zhang, Z. (1994). Estimating motion and structure from correspondences of line segments between two perspective images. Technical Report RR-2340, INRIA.

Appendix A

Imaging Geometry

Before discussing how 3D information can be obtained from images it is important to know how images are formed. First, the camera model is introduced, and then some important relationships between two views of a scene are presented.

A.1 The Camera Model

In this work the perspective camera model is used. The projection of an object point in the image is done in two steps. First the object is transformed in the camera reference frame and then projected into the image plane.

The relation between the coordinates of a point P in world $P_w = [X_w \;\; Y_w \;\; Z_w]^T$ and camera frame $P_c = [X_c \;\; Y_c \;\; Z_c]^T$ is:

$$P_c = R(P_w - T) \tag{A.1}$$

where R and T are the extrinsic parameters of the camera. R is the 3x3 rotation matrix and T is the 3x1 translation vector. This transformation has six degrees of freedom: three translation and three rotation components [Trucco and Verri, 1998].

Using the homogeneous representation of the points [Faugeras, 1995], equation (A.1) can be written as:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & -RT \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{A.2}$$

The matrix

$$M = \begin{bmatrix} R & -RT \\ 0 & 1 \end{bmatrix}$$

describes the object to frame transformation.

We denote by $p_c = [x_c \quad y_c \quad -f]^T$ the projection of P onto the image plane, expressed in the camera reference frame. f is the focal length of the camera. The equation of the perspective projection relating these points is:

$$\begin{bmatrix} x_c \\ y_c \\ -f \end{bmatrix} = \lambda \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{A.3}$$

where $\lambda = -f/Z_c$. This can be written as a linear mapping between homogeneous coordinates (the equation is only up to a scale factor):

$$\begin{bmatrix} x_c \\ y_c \\ -f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{A.4}$$

where the 3x4 projection matrix represents a map from 3D to 2D.

The point p_c is related to image coordinates $p_x = [u \quad v]^T$ through the following equations:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -f_x & -s & c_x \\ 0 & -f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = C \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \tag{A.5}$$

where $c = [c_x \quad c_y \quad 1]^T$ is the principal point, f_x and f_y are the focal lengths measured in width and height of the pixels, s is a factor accounting for the skewness due to non-rectangular pixels. These parameters are called the interior orientation parameters of the camera. The above upper triangular matrix is called the calibration matrix of the camera and C will be used to denote it.

Combining equations (A.2), (A.4) and (A.5) the following expression is obtained for a camera with some specific intrinsic and extrinsic orientation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -f_x & -s & c_x \\ 0 & -f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & -RT \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{A.6}$$

and can be simplified to:

$$p_x = C[R \mid \text{-}RT]P_w \tag{A.7}$$

The 3x4 matrix $P = C[R \mid \text{-}RT]$ is called projection matrix. This matrix is considered as known for our application. It can be be computed by the well-known calibration technique of Tsai [Tsai, 1987].

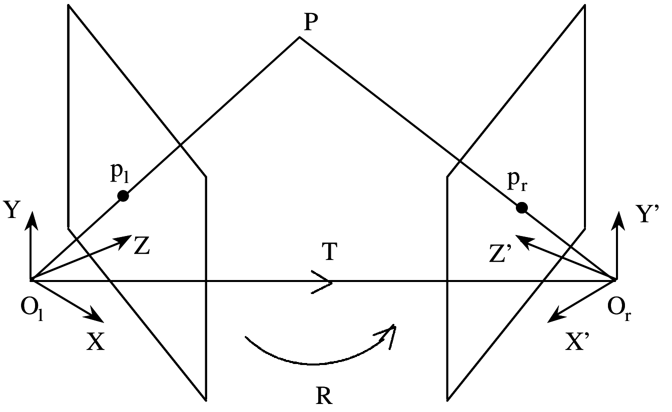


Figure A.1: Stereo configuration

A.2 Epipolar Geometry

Consider a point P_w in the world coordinate system (Figure A.1). The coordinates of this point in the left and right camera reference frame are $P_l = [X_l \;\; Y_l \;\; Z_l]^T$ and $P_r = [X_r \;\; Y_r \;\; Z_r]^T$ respectively. The vectors $p_l = [x_l \;\; y_l \;\; f_l]^T$ and $p_r = [x_r \;\; y_r \;\; f_r]^T$ refer to the projections of P onto the left and right image plane respectively, expressed in the corresponding camera reference frame. f_l and f_r are the focal lengths of the left and right camera. The points p_{xl} and p_{xr} are the points in pixel coordinates corresponding to p_l and p_r.

The projective geometry between two views is called epipolar geometry. The epipolar geometry is independent of the scene structure and only depends on the internal parameters of the cameras and their relative pose. This intrinsic geometry is expressed by either the essential matrix E or the fundamental matrix F. Then, the epipolar geometry is described and the expressions of the essential and fundamental matrices are derived.

The reference frames of the left and right cameras are related via the extrinsic parameters. These define a rigid transformation in 3D space, defined by a translation vector T and a rotation matrix R. Given a point P in space, the relation between P_l and P_r is therefore:

$$P_r = R(P_l - T) \tag{A.8}$$

The rotation R and the translation T can be expressed in function of the extrinsic parameters of two cameras. From (A.1) we have: $P_w = R_l P_l + T_l = R_r P_r + T_r$

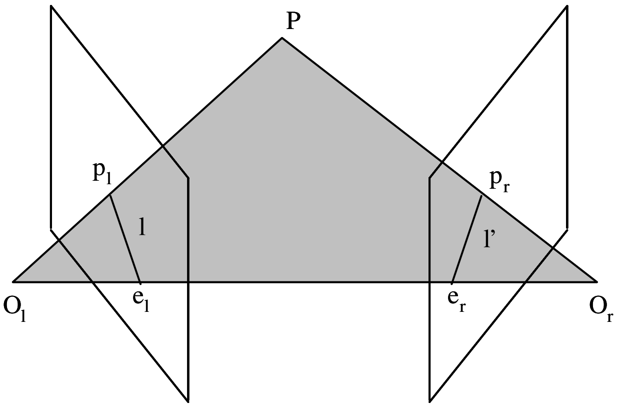


Figure A.2: Epipolar geometry

$R_l P_l + T_l = R_r P_r + T_r$

$P_r = R_r^T R_l P_l + R_r^T (T_l - T_r) = R(P_l - T)$

Hence

$$R = R_r^T R_l \quad \text{and} \quad T = R_l^T (T_r - T_l) \tag{A.9}$$

In Figure A.2 the point P in space is imaged in two views, at p_l in the first and p_r at the second. As shown in the Figure A.2 the image points p_l, and p_r, space point P, and the camera centers O_l and O_r are coplanar. The plane defined by them is called epipolar plane. The camera baseline $O_l O_r$ intersects the image planes at the epipoles e_l and e_r. Supposing that the point p_{xl} is known, the goal is to find how the corresponding point p_{xr} is constrained.

The epipolar plane is determined by the baseline $O_l O_r$ and the back projected ray from p_l. Hence, the ray corresponding to the undefined point p_r lies in the epipolar plane and it lies on the line of intersection l' of the epipolar plane with the second image plane. This line l', called epipolar line, is the projection in the second image of the ray from point p_l through the camera center O_l of the first camera. In terms of a stereo correspondence algorithm the benefit is that the search for the point corresponding to p_r does not need to cover the entire image plane but can be restricted to the epipolar line l'.

Thus, there is a mapping $p_x \mapsto l$ from a point in one image to its corresponding epipolar line in the other image. The goal is to express this mapping in terms of the parameters of the stereo system.

The equation of the epipolar plane through P can be written as the coplanarity condition of the vectors $\overrightarrow{O_lP_l}$, $\overrightarrow{O_rP_r}$ and $\overrightarrow{O_lO_r}$.

$$\overrightarrow{O_rP_r} \cdot [\overrightarrow{O_lO_r} \times \overrightarrow{O_lP_l}] = 0 \tag{A.10}$$

The coplanarity condition can be expressed in the coordinate frame associated to left camera as:

$$(R^T P_r)^T [T \times P_l] = 0 \tag{A.11}$$

Recalling that a vector product can be written as a multiplication by a rank-deficient matrix, we can write:

$$T \times P_l = SP_l \tag{A.12}$$

where

$$\mathbf{S} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \tag{A.13}$$

Using this fact (A.11) becomes:

$$P_r^T E P_l = 0 \tag{A.14}$$

with $E = RS$.

The matrix E is called the essential matrix and is the algebraic representation of epipolar geometry in camera coordinate systems. It establishes a link between the epipolar constraint and the extrinsic parameters of a stereo system. Using (A.3), (A.14) can be rewritten as:

$$p_r^T E p_l = 0 \tag{A.15}$$

which gives a relation between image points in the camera coordinate systems.

Since each image plane can be thought of as a subset of the projective space P_2, image points can be thought of as points of the projective plane P_2. Consequently, Ep_l can be thought as the projective line in the right plane u_r, that goes through p_r and the epipole e_r:

$$u_r = Ep_l \tag{A.16}$$

Hence, the essential matrix is the mapping between the points and epipolar lines we were looking for.

Then, using the intrinsic parameters to relate the image point in pixels to the point in the camera coordinate system:

$$p_l = M_l^{-1} p_{xl} \quad \text{and} \quad p_r = M_r^{-1} p_{xr} \tag{A.17}$$

and substituting in (A.15),

$$p_{xl}^T M_r^{-T} E M_l^{-1} p_{xl} = 0 \tag{A.18}$$

which defines the fundamental matrix F, by

$$p_{xr}^T F p_{xl} = 0 \tag{A.19}$$

where

$$F = M_r^{-T} E M_l^{-1} \tag{A.20}$$

Like the essential matrix in (A.16), Fp_{xl} can be thought as the equation of the projective line u_{xr} that corresponds to the point p_{xl}.

$$u_{xr} = Fp_{xl} \tag{A.21}$$

The fundamental matrix can be computed non-linearly from seven points, linearly from eight, and a least-squares solution can be found from more than eight [Hartley and Zisserman, 2001]. The main difference between the essential and the fundamental matrix is that the fundamental matrix is defined in terms of pixel coordinates, the essential matrix is defined in terms of camera coordinates.

Appendix B

Geometric Constraints

We express the planar coordinates of the corners of a rectangular based building model (Figure B.1) using the parameters of the model. The parameters of the ground plan of a rectangular building model are the x, y coordinates of the base point, the width (w), the length (l) and the orientation (k). Then the coordinates of the four corners are:

$$\begin{aligned}
c_0: \quad & x_0 = x \\
& y_0 = y \\
\\
c_1: \quad & x_1 = x + w\cos k \\
& y_1 = y + w\sin k \\
\\
c_2: \quad & x_2 = x + w\cos k - l\sin k \\
& y_2 = y + w\sin k + l\cos k \\
\\
c_3: \quad & x_3 = x - l\sin k \\
& y_3 = y + l\cos k
\end{aligned} \tag{B.1}$$

The relations specifying different constraints, except the parameter constraint, are expressed by nonlinear equations. In order to include these equations in the least-squares adjustment of the model fitting, the equations have to be linearized. For linearizing the relations we used a Taylor series expansion. The Taylor series expansion of a multivariate function $f(X)$ with n variables around the point $X_0 = (x_{10}, x_{20}, ...)$ is given by:

$$f(X) = f(X_0) + J(X_0)^T(X - X_0) \tag{B.2}$$

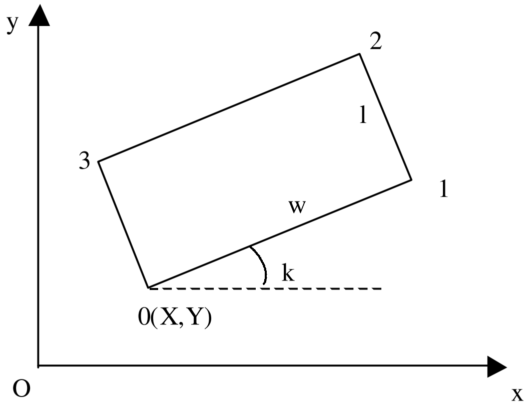


Figure B.1: Ground plan of a rectangular building primitive

where J is the Jacobian of the function f and it is defined as:

$$J(X) = \left[\frac{\partial f}{\partial x_1}(X) \quad \frac{\partial f}{\partial x_2}(X) \cdots \frac{\partial f}{\partial x_n}(X) \right]^T$$

B.1 Parameter Constraint

A parameter constraint establishes the relation between two parameters. It can be expressed as:

$$p - factor \cdot p' - offset = 0 \tag{B.3}$$

In order to include this relation into a least-squares adjustment that estimates the changes of the parameter values, we have to rewrite it as follows:

$$p - p_0 - factor(p' - p'_0) - offset = p_0 - factor \cdot p'0$$

or

$$\triangle p - factor \cdot \triangle p' = p_0 - factor \cdot p'_0 + offset \tag{B.4}$$

where p_0 and p'_0 are the initial values of the parameters p and p'.

B.2 Connection Constraint

Consider two neighboring building models belonging to the same building structure. The corners of the ground plan of the first building model are (c_0, c_1, c_2, c_3)

and the corners of the ground plan of the second building model are (c'_0, c'_1, c'_2, c'_3) (Figure B.2).

A connection constraint means that one edge of a building model lies on one of the edges of the other building model. There are four possible configurations for having a connection constraint between two building models. However, due to symmetry, only two of them have to be considered.

For the configuration from Figure B.2a, the connection constraint can be expressed as:

1. parameter constraint. The two building models have the same orientation: $k = k'$
2. corner c'_0 on line c_1c_2. This relation is fulfilled if the distance from corner c'_0 on line c_1c_2 is 0.
$$(x - x'_0)\tan k - y + y'_0 = 0$$
or
$$(x - x')\sin k + (-y + y')\cos k = 0 \tag{B.5}$$

This equation is linearized, resulting in the following equation:

$$\begin{aligned}\triangle x \sin k_0 - \triangle y \cos k_0 + \triangle k((x_0 - x'_0)\cos k_0 + (y_0 - y'_0)\sin k_0)\\ -\triangle x' \sin k_0 + \triangle y' \cos k_0\\ = -(x_0 - x'_0)\sin k_0 + (y_0 - y'_0)\cos k_0\end{aligned} \tag{B.6}$$

where x_0, y_0, k_0, x'_0 and y'_0 are the initial values of the parameters.

B.3 Corner Constraint

If two building models share a common corner, then there is a corner constraint between them. There are eight possible configurations from which only four have to be considered.

For the configuration from Figure B.2b, the corner constraint can be expressed as:

1. parameter constraint. The two building models have the same orientation: $k = k'$
2. corner c_1 is the same to the corner c'_0. This means that the x and y coordinates of the two corner points expressed in the same coordinate system are equal. So we have two relations:

$$\begin{aligned}x + w\cos k = x'\\ y + w\sin k = y'\end{aligned} \tag{B.7}$$

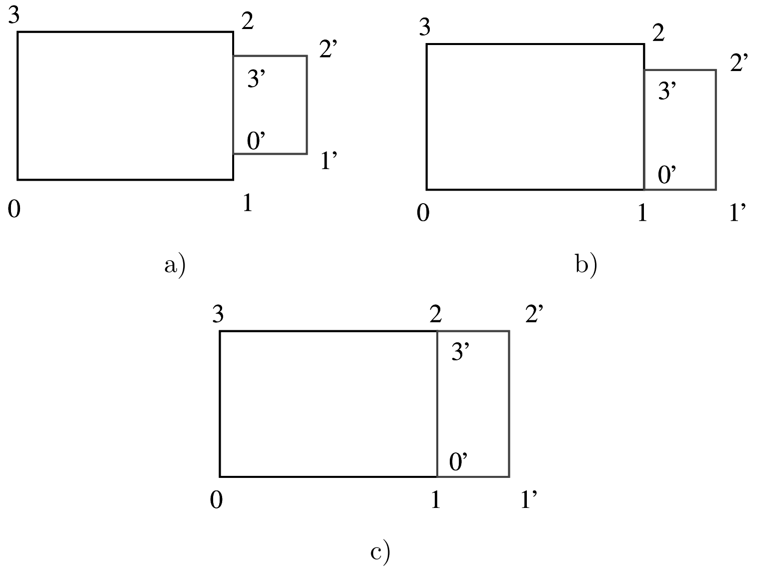


Figure B.2: Constraints. a) Connection constraint b) Corner constraint c) Extension constraint

By linearizing these equations we get:

$$\begin{aligned}
\triangle x + \triangle w \cos k_0 - \triangle k \cdot w_0 \sin k_0 - \triangle x' &= -x_0 - w_0 \cos k_0 + x'_0 \\
\triangle y + \triangle w \sin k_0 + \triangle k \cdot w_0 \cos k_0 - \triangle y' &= -y_0 - w'_0 \sin k'_0 + y'_0
\end{aligned} \tag{B.8}$$

B.4 Extension Constraint

If two building models share a common edge, then there is a extension constraint between them. There are four possible configurations from which only two have to be considered.

For the configuration from Figure B.2c, the extension constraint is expressed as:

1. parameter constraint. The two building models have the same orientation $k = k'$
2. corner constraint. The corner c_1 is the same to the corner c'_0.
3. parameter constraint. The two building models have the same width $w = w'$

Curriculum Vitae

Ildiko Suveg was born in Dej, Romania on October 11th, 1971. In 1990 she started to study Computer Science at Technical University of Cluj-Napoca, Romania. She received her Master's degree in 1995. The subject of her master thesis was "An Interactive System for Digital Image Processing".

Next year she continued her research on designing general purpose image processing systems at the Computer Science Department as a post-graduate student. This work resulted in the thesis entitled "Object Oriented Design of Image Processing and Analysis Systems".

In 1997 she spent one year at Technical University of Munich, Germany as a visiting scientist. Here she worked on realtime simulation of intelligent machines.

In October 1998, she started her Ph.D. research at the Photogrammetry and Remote Sensing Group of the Faculty of Civil Engineering, Delft University of Technology. The results of this work are presented in this thesis.