BUILDING RECONSTRUCTION
FROM TERRESTRIAL VIDEO IMAGE SEQUENCES

Yixiang Tian
June 2011

UNIVERSITY OF TWENTE
Faculty of Geo-information Science and Earth Observation

ITC dissertation number 186
ITC, P.O. Box 217, 7500 AE Enschede, The Netherlands

ISBN 978-90-6164-306-7
Cover designed by
Printed by ITC Printing Department

BUILDING RECONSTRUCTION
FROM TERRESTRIAL VIDEO IMAGE SEQUENCES

DISSERTATION

To obtain
the degree of doctor at the University of Twente,
on the authority of the Rector Magnificus,
prof.dr. H. Brinksma
on account of the decision of the graduation committee,
to be publicly defended
on Thursday June 9, 2011 at 16:46 Hrs

by

Yixiang Tian

born on February 24th , 1981

in Wuhan, China

This thesis is approved by
Prof. dr. ir. M. G. Vosselman, promotor
Prof. dr. Q. Zhu, promoter
Ass Prof. dr. M. Gerke, supervisor

Abstract

Realistic 3D city models are needed in fields like urban planning, virtual tourism, navigation and emergency response. For a number of applications, 3D city models are more interesting when they represent facades in great detail. These models have to be constructed from data recorded at street level.

Commonly, ground-based object extraction has mainly relied on manual operations with the support of image-based modeling software, such as Autodesk 3ds Max, Sketchup, and Autodesk ImageModeler. Due to the huge number of urban objects in a city and their variety of shapes, the manual reconstruction of a city is a time-consuming and expensive procedure. In recent years, image sequences have played an important role in many close-range applications in computer vision. Economic and flexible data acquisition procedures, together with the automatic structure from motion (SFM) approach, are advantages of using video image sequences as the data source for reconstructing objects. The high overlap of images in a video image sequence leads to a high redundancy of observations for each feature and there is only a slight difference between the observations of the same feature in neighboring conjugate images. Therefore, corresponding features in image space can be more easily identified by feature tracking than by feature matching from wide baseline images. However, there are some factors that prevent video data being used in photogrammetric applications that require high geometry accuracy. One is that video cameras are usually non-metric cameras, whose resolution is low and image quality worse than the quality provided by mapping cameras. Another problem is that the short baseline between the images leads to a poor ray intersection geometry and thus to a weak 3D position recovery across the sequence.

This PhD research aimed to reconstruct building facade models from terrestrial video image sequences using a largely automated process. Videos are recorded using a hand-held consumer camera or a camera mounted on a moving car. As the majority of buildings satisfy the assumption that they can be modeled geometrically as an ensemble of planar polygonal surfaces, using polyhedral models seems to be a relatively simple and efficient way to present building structures. Since

geometric models contain topological relations, presenting buildings in this way – rather than for instance a representation from a simple meshing of 3D points – enables one to perform a wide range of analyses in a CAD or GIS environment that need boundary and face representations for individual buildings. The knowledge of generic building structures can be implemented as rules and constraints which provide essential guidance for grouping extracted features to surfaces and then recovering building models. Using building structure knowledge therefore leads to a simple and reliable reconstruction method, and also enables one to obtain the main structures of buildings.

Starting from 3D points tracked from a video image sequence, the point accuracy is first analyzed to obtain reliably matched points. In order to introduce more constraints for the reconstruction and to fill the gaps in 3D point clouds, 3D edges are also used as primitives for the reconstruction. Extracted feature points and edges are grouped and verified according to predefined rules. After estimating plane parameters from all the edges and points in the plane, the knowledge about the generic shapes of building surfaces guides the generation of an outline. As the surface patch generation method is based on extracted features, the pure geometric reconstruction fails to determine some surfaces due to occlusions and imperfect feature extraction. In the model reconstruction step, these surface patches need to be correctly connected. Therefore, a hybrid model- and data-driven method is proposed to reconstruct a building model from both extracted surface patches and hypothesized parts in occluded areas. Topological constraints help us to make hypotheses and to define relations between different surface patches. Finally, textures are generated from image sequences and mapped on facades to provide a realistic visualization. The innovational aspect of this thesis work was to take advantage of redundant information from image sequences to improve the process of feature extraction and to generate hypotheses for surface patches. Another innovation was the modeling and use of knowledge about the structure of buildings in urban environments to improve the process of structure recovery.

Nine image sequences were used for testing the proposed building reconstruction method. Some of them only showed part of a building facade facing the street only, some were recorded around a building to capture all facades that were visible from the ground, and some were recorded to test the method's usability for rows of buildings in a street.

The reconstructed models were evaluated with respect to surface completeness, surface correctness, geometry accuracy and topological correctness. The results showed that this method correctly sets up topological relationships between generated surface patches and also obtains reasonable structure models in occluded areas. The average surface completeness was 94% and the surface correctness was 97%. The proposed method can successfully recover main facades within the geometry tolerance defined for LOD4 building models. The reconstructed models therefore satisfy the requirements for both visualization and analysis in most fields.

Acknowledgements

At the end of writing my PhD thesis, I am facing such mixed feelings, excitement, regret and relaxation. The period of the PhD study was a unique experience for me; it let me think about problems from a broader point of view and along many different directions within and beyond research.

Looking back at the past four years, there were many moments of happiness and sadness. This thesis could not have been finished without the help and support of many people who are all gratefully acknowledged here.

First of all, it is an honor for me express my deepest gratitude toward my supervisors (*promoters*), Prof. George Vosselman and Prof. Qing Zhu. They offered me valuable suggestions, constant guidance and criticism based on their profound knowledge and rich research experience. George, thank you for giving me the opportunity to get to know a new country, new culture and many interesting people. Prof. Zhu has been my supervisor since I was an MSc student and has provided help for all the aspects of my work.

I would like to express my gratitude to my supervisor, Markus Gerke. Thank you for your patient, constructive suggestions, timely assistance, and words of encouragements. I would also like to thank my supervisor Stephan Heuel, who proposed this topic and helped me at the beginning of my PhD period.

Of course, I am also indebted to many colleagues in the Department of Earth Observation Science at ITC and the VGE Group in Liesmars for inspiring discussions. I would like to thank all the people who helped me in capturing video data for my experiments. And I would also like to thank all my friends in the Netherlands, you stopped me from feeling lonely in a foreign country.

Lastly, but most important, I would like to thank my parents and my husband for their patience and support. Their encouragement turned my tears to smiles and made my PhD life much easier.

Table of Contents

1 Introduction

Applications such as Google Earth and Microsoft Bing maps (Virtual Earth) are very successful in delivering effective visualizations of the Earth's surface based on aerial and satellite images to a broad audience. However, various fields need realistic 3D city models, such as urban planning, virtual tourism, navigation and emergency response.

1.1 Motivation

There have been many attempts to document the spatial and visual complexities of real world environments. People use cameras or video cameras to record scenes, persons and objects of interest. Although much knowledge has been considered and most geometric relationships have been well established, it is still difficult to interpret 3D information correctly from 2D images by computers (Mayer, 2008). Some researchers concentrate on making visual 3D impressions as 'seen' from specific viewpoints (Shum and Kang, 2000). However, constraining the user to a predefined viewpoint means 3D models are limited to being used for visualization only, and further analysis or measurements cannot be done.

Depending on the applications, there are different requirements for 3D models, including high geometric accuracy, photo-realism of the results or complete details, as well as the requirements of the modeling technique, such as automation, real time, low cost and flexibility. Objects themselves can be rigid ones, such as buildings and roads, or non-rigid ones, such as human faces and bodies. Buildings are recognized as the most important objects in a city and the need for 3D virtual models of buildings has evolved rapidly in recent decades, because building models are now needed in many applications, such as navigation, urban planning, architectural design, heritage protection, object identification, and emergency response. However, there is a huge variety of buildings. Such a high flexibility and complexity increases the difficulty of modeling building structures by recovering 3D information from 2D images. Although much research is being devoted to building extraction and reconstruction, it is still far from the goal of a fully automated system. Recently, 3D city models constructed from ground-based data have become interesting as they represent realistic facades that contain more details than models constructed from aerial data.

Commonly, ground-based object extraction has mainly relied on manual operations with the support of modeling software, such as Autodesk 3ds Max, Sketchup, Autodesk ImageModeler or PhotoModeler (Autodesk, 2011b; Autodesk, 2011a; Eos Systems, 2011; Google, 2011). Manual reconstruction of a city is a time-consuming and expensive procedure due to the huge number of urban objects in a city and the variety of shapes (Brenner, 2005). Some semi-automated and fully automated reconstruction methods have been presented for use with different input data and aims (Debevec et al., 1996; Werner and Zisserman, 2002; Dick et al., 2004; Pollefeys et al., 2008; Pu and Vosselman, 2009). Given all these aspects, an automated, low cost and appropriate method is needed to reconstruct building models that can be used for describing the form of buildings. Building models mean the representation, not the actual description,

and these thus correspond to mathematical generalized models, which describe rules of a certain kind (Förstner, 1999). Since the majority of buildings satisfy the assumption that they can be modeled geometrically as an ensemble of planar polygonal surface patches, economic and flexible data acquisition procedures become possible and have been given much attention in recent years.

1.2 Characteristic of video image sequence

In recent years, the fields of photogrammetry and computer vision have grown towards each other. Computer vision focuses on the development of the algorithms for the automated extraction of information, while the photogrammetric algorithms focus more on obtaining the required information as precisely as possible (Förstner, 2002). Video image sequences are an important data source and have been analyzed by researchers in the computer vision field for decades. They have caught the attention of people with a photogrammetry background more and more in recent years. Economic and flexible data acquisition procedures, together with the automatic structure from motion (SFM) approach, are the main advantages of using video image sequences as the source data for object reconstruction.

There are several reasons why video image sequence is a reasonable data source for 3D object modeling:
1) low cost of equipment and simple acquisition process
2) fewer requirements for advanced operator skills compared with capturing other types of images for 3D modeling
3) fewer photometric changes in images, and
4) a high percentage of overlap between adjacent images.

These advantages are sufficient reason to consider video data in the conventional photogrammetric workflow. The high overlap of images in a video image sequence leads to a high redundancy of observations for each feature and there is only a slight difference between observations of the same feature in neighboring conjugate images. Corresponding features in image space can therefore be identified more easily by feature tracking than feature matching from wide baseline images. Video has been widely used for applications in the field of computer vision and in industrial applications, such as feature detection, feature tracking, navigation and 3D object modeling. However, there are some factors that prevent the use of video data in photogrammetric applications that require high geometry accuracy. One is that video cameras are usually non-metric cameras, whose resolution is low and image quality is worse than the quality provided by mapping cameras. There are few internal and external camera parameters provided for image sequence processing. Another problem is that the short baseline between the images leads to a poor ray intersection geometry and thus to a weak 3D position recovery across the sequence. Using video data in a wider range of applications will become possible if appropriate methods can be developed to overcome these limitations.

1.3 Scope and limitations

This research focuses on the modeling of buildings in the real world from terrestrial image sequences. The input video image sequences are recorded using a hand-held consumer camera or a camera mounted on a moving car. The research scope of this PhD research was limited to reconstructing building facades that are visible from terrestrial video image sequences. This means the building structures described here only refer to the exterior boundaries of the buildings not to their interior structures. Invisible parts of buildings are not directly recovered from the input data, although some of them may be recovered from hypotheses as a by-product. Just as with other data-driven methods, models in this research are reconstructed based on features extracted from images. If there are not enough features to support the existence of such buildings or facades, they cannot be reconstructed, but initial building detection is not the aim of this research work. Since this research is based on an assumption that the majority of buildings can be modeled geometrically as an ensemble of planar polygonal surface patches, buildings with planar structures are the target of this research, whereas those with curved structures are outside its scope.

The geometric reconstruction of buildings is the main aim of this research. For video image sequences, structure from motion technique, as the first step to recovering projective geometry for images, has long been studied. Its theory has been well defined and some software can provide acceptable results. The challenging part of this step is to improve the accuracy of the result, but this lies outside the scope of this research work. To make models appear visually realistic, texture mapping is an important step. However, it is not necessary for constructing geometrically and topologically correct 3D models, so no substantial research has been done in this direction.

Except for setting parameters and checking the corresponding result, all the processes described in this thesis are automated. This means 3D features and structures are reconstructed automatically and no human interpretation is involved. The human activities are restricted to ensuring that building facades can be observed in the captured image sequences and to selecting appropriate parameters in order to get good, intermediate or final results.

1.4 Research problems

3D building models with geometric and topologic information are useful for further interpretation and applications, but how to automatically recover the accurate structure of buildings in the real world is still an unsolved problem. A set of geometric features (points, edges or regions) without topologic connection is not enough for a complete 3D description and existing reconstruction algorithms lack a robust and comprehensive way to recover structures from geometric features. To find a solution to recovering building structures from geometric features was the main aim of this research. In this PhD research, the following problems and questions were considered.

- How to exploit the redundant information from image sequences and, at the same time, avoid the influence of a short baseline?

- What kinds of features should be chosen as primitives for modeling and how to deal with the problem of mismatching? Does the varied feature density for different parts of a building affect its model reconstruction?

- Using video as input data provides many images, 2D and 3D features during processing. At the same time, the geometry of objects can be described by different kinds of models. How to choose effective and flexible ways to present buildings?

- How to effectively integrate building facade or roof knowledge into the reconstruction process? What kind of constraints should be considered?

- Conflicting conditions may occur during the model reconstruction. How can these be avoided or solved?

- How to evaluate the quality of generalized building models, especially when there is no reference data?

1.5 Objective and innovation

The main objective of the work described here was to develop methods for recovering the structures of buildings from video image sequences that are captured by monocular cameras facing building facades.

This aim can be split into several aspects:

- To produce a geometrically and topologically correct 3D model. The final 3D models for presentation include both geometric and topological information about buildings. The algorithm should be capable of deciding which facades should be connected with a certain strategy, and which ones should be left out.

- To assess the importance of integrating building knowledge into the reconstruction. The algorithm should be capable of deciding which cases can be reconstructed with a certain level of knowledge.

- To build up a useful strategy for evaluating the quality of reconstructed models and to demonstrate the usability of video image sequences for 3D modeling of buildings at a certain level of detail.

The innovational aspects of this work were to take advantage of redundant information from image sequences to improve the processes of feature extraction and to generate hypotheses for surface patches. Another innovational aspect was to demonstrate how the structure of buildings in urban environments can be used to improve the processes of structure recovery.

1.6 Overview of the thesis

The next chapter introduces the state-of-the-art in building modeling. The first part of Chapter 2 gives an overview on automated and semi-automated building modeling methods, including the different data sources and model types. Methods for image-based building modeling are then focused, from low level feature extraction to high level model reconstruction and to the final texture mapping. The discussion in Section 2.6 leads on to Chapter 3, where a new approach to reconstructing building models from terrestrial video image sequences is presented. The basic outline and preprocessing steps are described in Chapter 3, followed by more details in Chapters 4 and 5. Chapter 4 focuses on feature extraction and feature grouping. The modeling of buildings and giving them texture is presented in Chapter 5. Chapter 6 consists of two sections: the first introduces the theories and functions used for evaluation, followed by appropriate aspects designed as evaluation factors. The second section reports on experiments and includes an evaluation of the results. In Chapter 7, the conclusions and outlook are presented.

2 State of the art in building modeling

This chapter gives an overview of the current research status into modeling techniques for buildings, based on a study of recent literature. Three-dimensional modeling of objects and scenes has been a topic of intense effort for many years by computer vision and photogrammetric communities. It can be seen as the complete process that starts from data acquisition and ends with a 3D virtual model that may be visually interactive on a computer (Remondino and El-Hakim, 2006; Mayer, 2008).

In general, there are two common ways to capture data for 3D modeling of real objects and scenes. One is based on active range data (e.g. structured light, laser scanning), and the other is based on camera or video images. Active range-based modeling methods directly capture the 3D geometric information of an object. They provide a highly detailed and accurate representation of shapes. On the other hand, they can also be cumbersome and expensive (Brenner, 2000; Vosselman and Dijkman, 2001; Tarsha-Kurdi et al., 2007; Barber et al., 2008; Becker et al., 2008; Pu and Vosselman, 2009). Image-based modeling is widely used for obtaining virtual models of objects. As the requirements of 3D models need to be specified for many different applications, image-based modeling may well have different meanings or focuses for different points. Some of the terms used in this thesis are explained in the glossary. Since the main task in this research project was to reconstruct building models from video image sequences, only image-based methods are discussed and reviewed here.

Extraction and reconstruction of man-made structures from aerial images has been a topic of intense research for many years (Remondino and El-Hakim, 2006). Since only roofs can be well observed in aerial images, researchers focus on recovering roof structure from features (points, edges or segments) with or without other data, such as building ground plane (Henricsson and Baltsavias, 1997; Baillard and Zisserman, 1999; Suveg and Vosselman, 2004). Although a lot of research is still being devoted to this topic, interpreting 3D structures from 2D images is still far from the goal of a fully automatic system. Recently, 3D city models constructed from terrestrial data have become interesting (Debevec et al., 1996; Werner and Zisserman, 2002; Dick et al., 2004; Pollefeys et al., 2008): these close-range data include long baseline images, image sequences or videos. Researchers are addressing the problem of reconstruction relying solely on hand-held cameras in order to increase the flexibility of the system while reducing the size of equipment, its weight and cost. In recent years, some researchers have combine range-based and image-based approaches, and also included other data, e.g. 2D map data (Suveg and Vosselman, 2004; El-Hakim et al., 2005b; Zhang et al., 2005). Successfully combining different kinds of data together can improve the reconstruction process, especially in terms of time and reliability.

The main task of reconstruction is to rebuild geometric and topological relations within buildings; this is the most important aspect of building modeling. After a general introduction to some aspects of image-based building modeling in this section, the basic techniques of image processing that are useful for this research are described (sections 2.1 and 2.2). The means for presenting building models are discussed in section 2.3.

Section 2.4 focuses on building reconstruction methods from optical imagery, especially from video image sequences. Finally, texture mapping is described (section 2.5).

2.1 Feature extraction

How to extract features is the basic problem in photogrammetry and computer vision. This results in 2D or 3D features and the extraction problem needs to be solved when dealing with object tracking, object detection and 3D model reconstruction. Usually, 3D feature extraction includes three conjunctive steps: detection, matching and 3D coordinate computation. Detection aims to identify the locations of 2D features in each image. Feature-based matching (FBM) can be defined as establishing the correspondence between detected features. Despite a large number of studies, matching has long been, and still is, one of the most challenging tasks in photogrammetric research (Heipke, 1996; Jazayeri and Fraser, 2010). Features should be distinct with respect to their neighborhood, invariant with respect to geometric and radiometric influences, stable with respect to noise, but seldom with respect to other features (Förstner, 1986). In FBM, features are extracted in each image individually prior to matching them. Local features are points, edges or lines, and small regions. Global features comprise polygons and more complex descriptions of the image content called structures (Heipke, 1996). Global features are rarer and thus provide a better basis for reliable matching. However, it is difficult to define and extract global features, and they tend to be more application-dependent than local features. Therefore, points and edges are focused on in this research.

2.1.1 Point detection and matching

There are several commonly used interest point detectors in photogrammetry and computer vision, such as the Moravec detector (Moravec, 1977), Förstner detector (Förstner, 1986), SUSAN detector (Smith and Brady, 1997), Harris detector (Harris and Stephens, 1988), Harris-Laplace detector (Mikolajczyk and Schmid, 2004) and FAST detector (Rosten and Drummond, 2006). During the point matching, information surrounded interest points are used and some point descriptors have been proposed, such as the SIFT (Scale-invariant feature transform) operator (Lowe, 1999; Lowe, 2004) and SURF (Speed up robust features) operator (Bay et al., 2008). The SIFT still seems to be the most appealing descriptor for practical uses, and hence also the most widely used nowadays. More details about interest point operators and a comparison of them can be found in the references (Schmid et al., 2000; Tissainayagam and Suter, 2004; Remondino, 2006; Jazayeri and Fraser, 2010). It is obvious that each operator has its own advantages and limitations and that no single algorithm has been accepted as the best choice for all applications.

To achieve a reliable matching, some constraints are used that can be divided into geometrical constraints, similarity constraints and compatibility constraints. The most common for geometry is the epipolar constraint (Ackermann and Krzystek, 1995), while grid or Delaunay triangle constraints also limit matching to a local area (Cross et al., 1997; Tang et al., 2002; Zhu et al., 2005). The similarity between two point features is usually evaluated by their gray values. Normalized cross correlation is a common way

to charge two pixels by a temple window (McGlone et al., 2004). The compatibility constraints include some prior knowledge for matching, e.g. a smoothness constraint, uniqueness constraint and ordering constraint (Marr and Poggio, 1979; Mayhew and Frisby, 1981; Grimson, 1985).

An image-matching algorithm consists of a number of steps, and the matching strategy is the most important aspect for such a method (Gülch, 1991; Faugeras et al., 1992a). An image pyramid is a coarse-to-fine hierarchy strategy, which reduces the searching area from a coarse to a fine resolution (Burt, 1988; Marapane and Trivedi, 1994; Hung et al., 1998). Multiple images provide redundant information to avoid errors by least squares estimation (Grün and Baltsavias, 1988; Helava, 1988; Okutomi and Kanade, 1993; Maas and Kersten, 1997; Tao, 2000). There are also some integration strategies that include compatibility constraints in the matching strategies. Dynamic programming, relaxation and artificial neural networks are widely used in image matching (Barnard and Thompson, 1980; Zhang et al., 2000; Jiang et al., 2003; Galo and Tozzi, 2004). However, there is no method that can guarantee all the matching results are correct.

Since video streams are acquired at a high frequency, frame-to-frame differences are small enough to let interest points be tracked though the image sequences. Point detection and matching steps are therefore done through feature tracking in computer vision (Heinrichs et al., 2008). The most widely used tracker is the Kanade-Lucas-Tomasi (KLT) tracker (Lucas and Kanade, 1981; Shi and Tomasi, 1994). Some improvements or implementations have been made to the KLT (Bouguet, 2000; Sinha et al., 2007), e.g. the iterative algorithm (Birchfield, 2011) computes the optical flow of interest points using image pyramids. Errors can accumulate during the tracking of features over many frames. To detect bad matches, features in the current frame is compared to features in the first frame, but due to distortion of perspective, intensity-based consistency checks must be performed with an affine mapping (see Fig. 2.1).

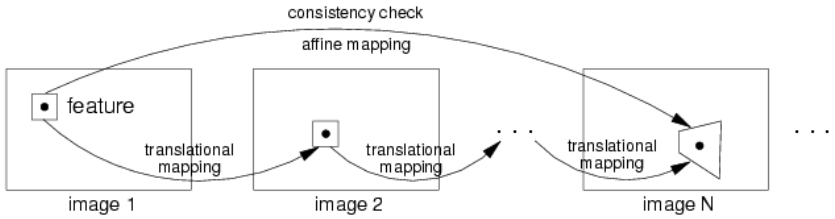


Figure 2.1 Affine consistency checking of feature matching (Birchfield, 2011)

By determining 2D-2D point correspondences in consecutive video frames, the relative camera geometry can be computed through bundle adjustment. Further information on structure from motion (SFM) can be found in section 2.4.

2.1.2 Edge detection and matching

Using edge information for reconstructing man-made objects from images has been performed by researchers from the fields of photogrammetry and computer vision for a

long time (Hartley and Zisserman, 2004). Edge features can provide more constraints for an object's shape than point features.

2.1.2.1 Edge detection

Edge detection is a term used in image processing and computer vision, particularly in the areas of feature detection and extraction; it refers to algorithms which aim to identify optical edges that correspond to discontinuities in the physical, photometrical and geometrical properties of scene objects (Ziou and Tabbone, 1998; Ando, 2000). In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicate the locations of these optical edges. However, it is not always possible to obtain such ideal edges from images since image derivatives are sensitive to various sources of noise (Ziou and Tabbone, 1998).

A widely used edge detection algorithm is the Canny edge detector, which focuses on an ideal step edge and can attain sub-pixel accuracy (Canny, 1986). The Canny operator detects edges in three steps. Firstly, image noise is eliminated by smoothing the image using a Gaussian kernel. The operator then highlights regions with high spatial derivatives according to the image gradient. Finally, the algorithm tracks along these regions and sets pixels to zero that are not at the maximum. The gradient array is further reduced by the upper and lower thresholds.

There are many improving methods for using the Canny edge detector, such as the EDISON edge detector (Meer and Georgescu, 2001), in which a confidence measure is introduced and integrated into gradient-based edge detectors. By defining two decision boundaries $f^{(Lower)}(\rho,\eta)=0$ and $f^{(High)}(\rho,\eta)=0$, then, according to its estimated gradient magnitude (ρ) and its confidence value (η), each pixel can be labeled as an edge or not.

2.1.2.2 Straight line extraction

Straight lines are important clues for visual perception and the fundamental basis of image interpretation, especially when interpreting man-made objects, such as buildings (Zhang et al., 2004). After edge detection, boundary points are detected and they need to be linked together to form straight lines.

The Hough transform is a technique which can be used to isolate features of a particular shape within an image by a voting procedure (Hough, 1962). In the image space, the straight line can be described as:

$$y = ax + b \tag{2.1}$$

where a is the slope and b is the intercept.

It can also be represented as the distance (ρ) between the line and the origin and the angle (θ) of the vector from the origin to this closest point:

$$\rho = x\cos\theta + y\sin\theta \tag{2.2}$$

An image point (x,y) corresponds to a curve in the parameter space. By considering all the points on the line, the point where all the curves intersect indicates the correct (ρ,θ) for that line.

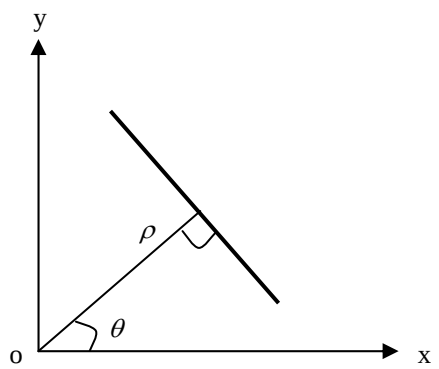


Figure 2.2 Parametric description of a straight line

The Hough transform is the typical, indirect straight-line extraction method, which turns the global detection problem in the image space into a local peak detection problem in a transform space. There are some improved methods based on it, such as the Fast Hough transform (Li et al., 1986), Adaptive Hough transform (Illingworth and Kittler, 1987), Multi-resolution Hough transform (Atiquzzaman, 1992) and Fuzzy Hough transform (Soodamani and Liu, 1998). Warrick and Delaney (1997) provided a method that extracts straight lines based on a random transform with a wavelet filter. Since the Hough transform is a global process, the lines are formed by the whole image information. There are some methods that directly extract lines based on local gradient or intensity information, e.g. the Heuristic linking method (Nevatia and Babu, 1979), Burns segment detector (Burns et al., 1986), Token-based extraction (Boldt et al., 1989), Line space-based extraction (Zhang et al., 2004) and Robust approach (Wen and Wang, 2001). More details and a review of straight-line extraction methods can be found in (Xu, 2007).

2.1.2.3 Edge matching

There are many factors that interfere with a successful edge matching. One is that edges belonging to the same entity in object space are often extracted incompletely and/or inaccurately from a single image. Sometimes, an ideal edge might be broken into two or more small segments that are not connected to each other. Furthermore the endpoints may not be reliable, and even with a correct orientation, it is difficult to build up topological connections between edges. A second factor causing complexity in edge matching is the lack of strong, disambiguous geometric constraints available over more than three views during edge matching. There is only a weak overlap constraint for edge segmentations of finite length arising from applying the epipolar geometry constraint to endpoints (Schmid and Zisserman, 1997; Baillard et al., 1999).

Existing approaches to edge matching described in the literature are generally categorized into two types. One is matching individual edges between images based on

a similarity measure. The similarity measure is based on comparing edge attributes, such as orientation or edge support region information. The other strategy is structural matching, which considers more geometrical and topological information among the edge features. Those methods are highly complex and sensitive to errors in the segmentation process (Armstrong and Zisserman, 1995; Baillard et al., 1999; Kunii and Chikatsu, 2004; Klein and Murray, 2006). When camera projection information is available, matching individual edges can yield precise and efficient results, and reduce the complexity and computation time. Depending on the number of images used in matching, edge matching methods can be divided into those based on two views, three views or multiple views.

- Edge matching over two views

The epipolar constraint is widely applied to finite edge matching by generating two epipolar lines from two endpoints in the other image. Edges that intersect or that are contained by the region defined by these two epipolar lines are selected as candidates. Then the average of the correlation scores for edge pixels are calculated as the matching score for pair edges. There are some edge matching methods for general stereo matching, such as probabilistic relaxation (Rosenfeld et al., 1976), adaptive least squares correlation (Grün, 1985), and area correlation (Schenk, 1999).

- Edge matching over three views

The trifocal tensor extends edge matching from two views to three views, and uses the third view to verify a matching result from two views (Baillard et al., 1999). Kunii and Chikatsu (2004) described a method using optical flow and a trifocal tensor. However, the procedure cannot be applied for all edges due to fragments that lead to mismatching.

- Edge matching over multiple views

Most multiple edge-matching methods are based on stereo or triplet image pairs, and then the results are merged together or the initial results are verified in other images to reduce matching error (Schmid and Zisserman, 1997; Fitzgibbon and Zisserman, 1998; Baillard et al., 1999; Zhang et al., 2005). Using multiple images can yield redundant information for verification, and this method therefore overcomes the deficiencies of edge detection and increases the accuracy of the estimated 3D line.

2.2 Structure from motion

The structure (or shape) from motion problem, i.e. how to recover scene geometry and camera motion from a sequence of images, has attracted much attention in the computer vision community over years. Similar to the matching problem in photogrammetry, early researchers computed the structure and motion from a small set of points matched in two frames (Ullman, 1979; Longuet-Higgins, 1981; Tsai and Huang, 1984). Later research focused on longer image sequences, include both rigid and non-rigid objects (Tomasi and Kanade, 1992; Szeliski and Kang, 1994; Theobalt et al., 2002; D'Apuzzo, 2003; Torresani and Hertzmann, 2004).

In contrast to conventional (airborne) remote sensing, uncalibrated, non-metric cameras are used and precise navigational information from GPS/IMU is normally not available during the capturing of terrestrial image sequences. There is therefore no prior knowledge of camera poses or interior orientation parameters available. All parameters, as well as 3D coordinates, have to be estimated from the image sequence. The general steps include automatically extracting points of interest (such as corners), to sequentially match or track them across views and then to compute the camera parameters and 3D coordinates of the matched points through a subsequent bundle adjustment including self-calibration. The scene can be reconstructed up to scale if there is additional knowledge on the scene geometry available (Pollefeys et al., 2000; Hartley and Zisserman, 2004; Nistér, 2004a). Below, the self-calibration step, which is necessary for uncalibrated image sequences, will be introduced first.

Camera calibration refers to the determination of the parameters describing the internal geometry of the individual imaging devices and other parameters modeling the systematic errors caused by the optical system. Camera orientation includes the determination of the parameters of exterior orientation to define the camera station and camera axis in 3D space (D'Apuzzo, 2003). System calibration is an essential step in recovering the transformation between world coordinates and image coordinates. According to the dimensionality of the equipment employed, the existing camera calibration algorithms can be classified roughly into four categories: self-calibration, which is considered as 0D since no physical calibration objects are used; one-dimensional (1D) segment-based calibration, 2D plane-based calibration, and 3D object-based calibration. For rigid objects, a camera or cameras are often facing a turntable at an oblique angle with the whole calibration pattern in their field of view. The approach of (Tomasi and Kanade, 1992) uses an affine factorization method to extract 3D objects from image sequences. An important restriction of this system is the assumption of orthographic projection. Using additional camera calibration devices guarantees a precise calibration, but limits the flexibility in real scene applications (Koch et al., 2000). As self-calibration can recover the camera's interior orientation parameters by using information only contained in uncalibrated images, it is more useful in real scenes that have no prior knowledge or additional calibration equipment available. The theory of camera self-calibration was described in (Faugeras et al., 1992b). The radial distortion parameters are estimated as additional unknowns. The distortion can also be assessed before feature tracking by warping the original image.

For uncalibrated image sequences, camera calibration, camera orientation and feature tracking are usually combined together. Only the relative position of the cameras can be recovered, but the overall scale of the configuration can never be recovered solely from images without prior information on the scene. Based on projective geometry, the first two images are generally used to initialize the sequence. The final result is obtained by tracking salient image features throughout the sequence (Hartley, 1992; Nistér, 2004a; Pollefeys, 2004). A restricted number of corresponding points is sufficient to calibrate and orientate the camera. However, these matched feature points affect the precision of the camera calibration (Koch et al., 2000). If the set of matches is contaminated with even a small set of outliers, the result can become unusable. The approach that is used to cope with this problem is the RANSAC algorithm (Fisher and Bolles, 1981); an efficient solution to the classical, five-point, relative pose problem based on RANSAC

is presented in (Nistér, 2004b). To avoid the errors of point-feature tracking, some methods solve for shape globally over an entire sequence, and analyze optical flow for all the points in the image (Irani, 2002; Torresani and Hertzmann, 2004). Other approaches use different features, such as lines or curves, since these are more appropriate in real-world environments where there are few straight lines or edges that do not change radically with aspect and illumination (Clarke et al., 1996; Bartoli et al., 2003; Wilczkowiak et al., 2005b; Zhang et al., 2005). However, automatic edge tracking alone is not reliable, so most of the time, predefined 3D edges or manually matched edges are used as the initial value for edge tracking to estimate the camera's parameters.

The process after feature tracking can be generalized as follows:

a) estimate the initial camera parameters,
b) create a single 3D point for every tracked image point, and remove those with a high re-projection error, and
c) optimize the remaining points and the camera information parameters by minimizing all the re-projection errors of the 3D point positions.

In order to get generated elements to appear in the appropriate position and with the appropriate scale and orientation, it should be possible to use survey points or some reference distance measures for objects in the world coordinate system in the method used (Dobbert, 2005).

2.3 Building models

Some important building model types are discussed briefly in this section, since the way of presenting buildings can have an impact on the manner of building reconstruction and on further applications based on the model parameters (Vosselman, 1998).

If buildings are described by triangular network (TIN), which is called 'mesh' in computer vision community, the 3D geometry constructed from the result of triangulating the dense point clouds is very useful for recovering details. If the points are dense enough, the modeling process is almost the same as mesh simplification. However, it is badly affected by depth errors. In fact, for ordinary buildings, a small number of points are enough to present buildings with planar or cylindrical patches. A mesh does not use any prior object knowledge at all or exploit the fact that buildings mostly consist of simple planar faces.

The geometry of objects can be described by boundary representation (B-rep), constructive solid geometry (CSG) or spatial enumeration (i.e. voxels) (Brenner et al., 2003). CSG is widely used for computer-aided design (CAD). Using geometrical techniques, an object is composed by taking unions and intersections of several primitive shapes like rectangular boxes, spheres, cylinders, cones and tetrahedrons. Such an object model is described by specifying the values of the shape parameters and six pose parameters for each primitive. Often the absolute pose parameters are specified for one primitive only and the pose parameters of the other primitives are described relative to the first primitive. Boundary presentations of objects describe the geometry

of the points, edges and surfaces of the object boundaries together with the topological relations between these points, lines and surfaces.

Brunn and Weidner (1997) distinguished three kinds of building models: parametric, prismatic or polyhedral. Parametric models are used for simple buildings, which can be described using a few parameters. As a basis, assumptions have been made that the buildings are separate from each other and that the ground plan of the building is a rectangle. Complex buildings and blocks of buildings are described using prismatic models, which constitute the second group. These models are based on generic knowledge of the buildings, e.g. that the ground plans of buildings or building blocks are sets of closed polygons. Furthermore, neighboring straight lines of the buildings' outlines and therefore neighboring edges of the polygons are likely to be orthogonal. The outline of a building may also be formed by several polygons, e.g. representing court yards. In cases where parametric and prismatic models fail, Brunn and Weidber (1997) introduced polyhedral models.

CSG models, in contrast to polyhedral models, are able to represent parts of the buildings according to clear semantics. However, polyhedral models can describe structures with topological relations and are easy to use for further detailed reconstruction or in combining detailed models. Buildings are considered as any polyhedral surface with no overhang, whose external border is made up of vertical planes. This definition remains very general and can represent almost any building seen from aerial images in urban areas, with the exception of curved buildings (Heuel and Förstner, 2001; Taillandier and Deriche, 2004). As the majority of buildings satisfy the assumption that they can be modeled geometrically as an ensemble of planar polygonal surface patches, using polyhedral models seems to be a relatively simple and efficient way to represent building structures (Werner and Zisserman, 2002). Such representations, with detailed roof structures and planar facades, are sufficient for simulations or visualizations on a small- or medium scale, for example, and they satisfy the requirements of LOD2 and LOD3 as defined by CityGML (Open Geospatial Consortium, 2008).

2.4 Building reconstruction from optical imagery

According to the number of images used in a reconstruction, the methods can be divided into those based on a single image, stereo images and multiple images. Most approaches have focused on reconstructing specific building models: rectilinear shapes, flat roofs or parametric models. Recently, more generic reconstruction approaches have been proposed. In this section, an overview of contributions to the field of automatic and semi-automatic building reconstruction from images is given. Automatic methods can be identified as data-driven or model-driven.

2.4.1 Semi-automatic building reconstruction

Semi-automatic approaches are commonly used for building reconstruction, in particular in the case of complex architectures (Debevec et al., 1996; Englert and Gülch, 1996; Van den Heuvel, 1999; Sinha et al., 2008). Firstly, a set of 3D parametric models,

such as cuboids, prisms and pyramids are predefined. Then the operator selects an appropriate primitive to model the building, or part of the building, by pinning model vertexes or edges to specific locations in the different images. Complex buildings are modeled as combinations and variations of the basic primitives. Although single-view reconstruction is sometimes possible, multiple images make the results more reliable. Considering how laborious this work can become with more images, a few well-planned images are normally used. Blocks can also be used as primitives. El-Hakim et al. (2005a) proposed a method that reused building blocks to add details on coarse building models. However, all of these techniques are restricted to the use of solid primitives and cannot make use of a huge number of images in an easy and flexible way. Instead of using a set of predefined structures, Van den Heuvel (1999) proposed a method for specifying straight lines in images. Polyhedral building models can be recovered based on coplanar constraint, even when some vertexes are occluded. Sinha et al. (2008) let users draw outlines overlaid on images and utilize vanishing points calculated from them be the constraint used at multiple stages during the reconstruction. Buildings with planar polygonal surfaces can therefore be recovered by connecting neighboring surfaces.

Semi-automatic methods combine human image understanding and interpretation with the ability of computers, which provide reliable and useful ways of reconstructing buildings. There are some software programs available, such as Autodesk ImageModeler and PhotoModeler (Autodesk, 2011b; Eos Systems, 2011). Human operators play an important role in these systems and affect their efficiency and quality of result. Automatic extraction of objects from images is not only scientifically challenging, but also of major practical importance for data acquisition and updating geographic information system (GIS) databases or site models.

2.4.2 Data-driven versus model-driven building reconstruction

Automated building reconstruction methods can be divided into data-driven and model-driven approaches. The major difference between them is how they recognize building structures: in a bottom-up or top-down manner.

In the data-driven approaches, some kinds of primitives are used to reconstruct buildings, such as 3D segments in (Baillard and Zisserman, 1999; Scholze et al., 2002), corners in (Heuel and Förstner, 2001; Hilton, 2005) and planar patches in (Ameri and Fritsch, 2000; Taillandier and Deriche, 2004). Occlusions of buildings or building parts cause failure in the complete feature extraction. Therefore, some unreconstructed areas must be left or only the actually observed/detected features are connected if only a data-driven method is used. On the other hand, model-driven approaches (Zisserman et al., 2001; Dick et al., 2004) use models of buildings to restrict the set of possible shapes. Reliable identifications can be made by identifying consistent partial matches between the models and features extracted from the images, thereby allowing the system to make inferences about the scene that go beyond what is explicitly available from the images, such as lack of detection or over-detection. Model-driven approaches are providing promising results, because they link structural knowledge of buildings with features and find best-fit models. However, they are still limited to the prototypes used and thus

cannot handle all the shapes now found in urban or suburban areas. Besides, the robustness of the approaches lies intrinsically in the small number of models. Increasing the library of models would result in an increased complexity and reduced robustness. There are also some researchers combining probabilistic geometric and semantic reasoning and they are getting encouraging reconstruction results (Scholze et al., 2002).

An improved method can integrate structural knowledge into the reconstruction in order to deduce reasonable hypotheses (Baltsavias, 2004). Model-driven methods ensure the plausibility and topological correctness of the reconstructed objects. On the other hand, the enormous variation in the structure and shape of building facades prevents the use of too tight constraints to recover the structure. The geometric constraints included in the defined models should be used wherever possible (Brenner, 2005). A promising alternative seems to take advantage of the flexibility of a data-driven approach and the robustness of a model-driven one.

2.4.3 Terrestrial image sequences or video based

Using multiple images can reduce feature extraction errors, provide more validation and then increase the accuracy of the reconstructed models. Image sequences or video-based methods, which are of interest in research studies, as well as the development of multiple image-based methods, are highlighted below.

In this section, terrestrial image sequences or video-based building reconstruction methods are analyzed in detail. In recent years, this topic has received much attention (Fitzgibbon and Zisserman, 1998; Nistér, 2004a; Akbarzadeh et al., 2006; Mayer and Reznik, 2007; Cornelis et al., 2008; Pollefeys et al., 2008; Snavely et al., 2008). Economic and flexible data acquisition procedures, together with the automatic structure from motion approach, are the main advantages of using video image sequences as the source data for building reconstruction. The structure from motion technique has already been described in section 2.2. However, while promising, the widely reported fully automated methods are not always successful in practical applications. The first problem is that they rely on feature tracking results (as discussed in section 2.1.1). However, a sparse set of matched feature points is not enough for a complete 3D description.

1) Semi-automatic methods

Some approaches start from an approximate 3D model and camera poses and refine the model based on images (Debevec et al., 1996; Dick et al., 2004). The advantage is that fewer images are required, although preliminary models of a sufficient quality are often not available.

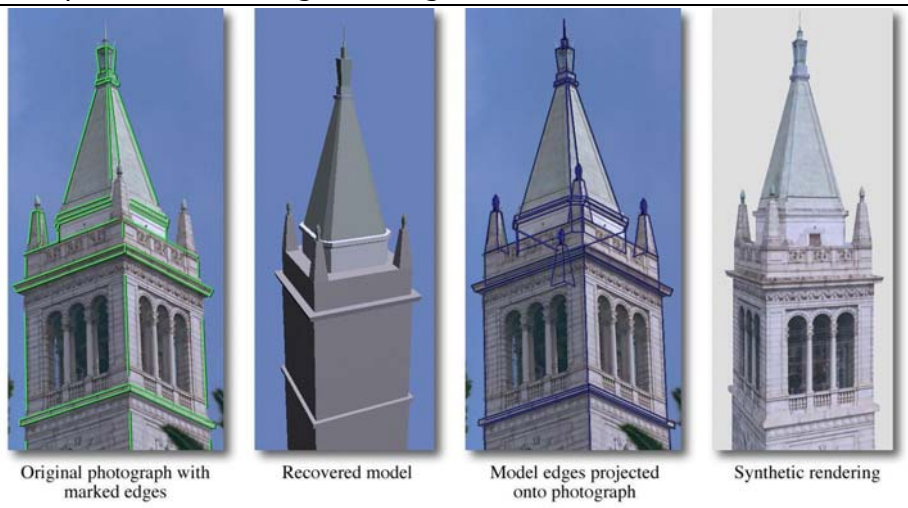


Figure 2.3 Modeling and rendering of architecture from photographs (Debevec et al., 1996)

2) Automatic methods with dense matching

Koch et al. (2000) introduced a 3-step surface description method. First, feature tracking is used to obtain camera pose and interior orientation parameters. Next, adjacent images are treated as stereo image pairs to compute depth maps by area matching. Finally, dense depth maps are fused together by linking all correspondences from all the images. The model is stored as a textured triangular surface mesh. Based on this research, multiple image sequences with GPS and IMU information can be collected in order to place the reconstructed models in geo-registered coordinates (Akbarzadeh et al., 2006; Pollefeys et al., 2008). Besides high quality in terms of both geometry and appearance, the model aimed at real-time performance, although additional data (GPS/IMU) are not always easily obtainable.

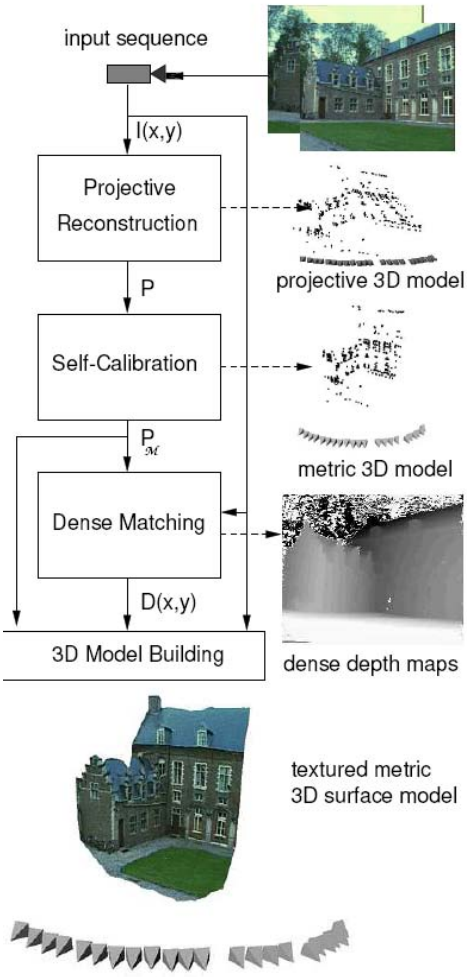


Figure 2.4 System overview from (Koch et al., 2000)

Fitzgibbon and Zisserman (1998) described a 3D structure recovering method by matching points and lines between triple images, and then merging triplets together. Dense points and edges have to be extracted from images. The limitation of this method is that it depends on the quality of images and texture of buildings. Dense 3D reconstruction from multiple image sequences was proposed in (Sato et al., 2003). First, the method estimates extrinsic camera parameters of each image sequence, and then it reconstructs a dense 3D model of a scene using extended, multi-baseline stereo and voxel-voting techniques to decide the position of a 3D point from several candidates. The global geometric constraint is used to avoid the problem of mismatching (Mukunoki et al., 2005). However, using only dense points without shape information cannot build up the structure of buildings and the dense depth map may not always be accurate enough to recover all the details of a building. Another problem for these

systems is that the required computation time is increasing as demand for details increases.

3) Automatic methods without dense matching

There also some studies that aimed at the interpretation of building facades from image sequences. Mayer and Reznik (2006) considered windows and doors to be basic parts in constructing the building facades. Based on sparse 3D points and edges recovered from images, buildings can be shown as textured 3D models, and compared with images from different periods for change detection (Schindler et al., 2007). Some systems use sideways-looking video to create multi-perspective images for visualizing roughly planar scenes, or they employ a simple model for geometry (Roman et al., 2004; Cornelis et al., 2008). However, these approaches aim at a visualization of buildings, and do not explicitly recover a boundary and face representation for individual buildings. Thus, analyses of buildings, for instance in a CAD or GIS environment, is not possible.

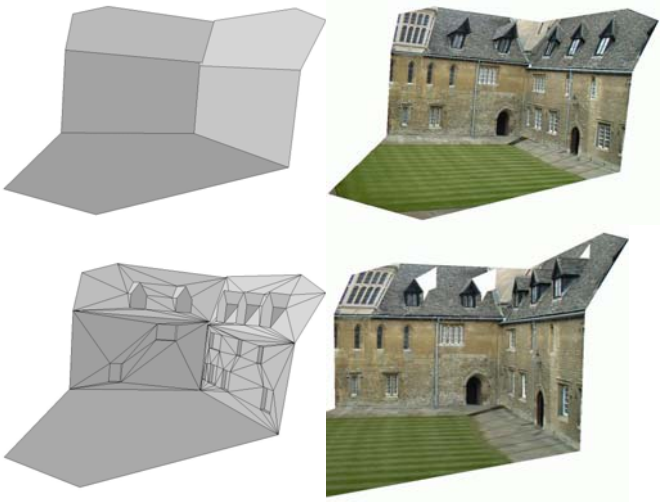


Figure 2.5 Coarse and final models from (Werner and Zisserman, 2002)

Werner and Zisserman (2002) proposed an approach that uses sparse points and edges to reconstruct ground- and building planes from wide baseline images. From a vanishing point computation, three principal directions are reconstructed. They sweep vertical planes through space to determine the position which best matches the images. Then two generic models are used to fit some details. However, the two orthogonal horizontal directions that are required in their method are not usually available from video image sequences.

An overview of recent terrestrial image sequence- or video-based projects is given in Table 2.1. Automatic image sequence- or video-based methods are still mainly used for visualization or navigation due to the limitations in geometric accuracy.

Table 2.1 Overview of terrestrial image sequence- or video-based projects

Project	**Data**	**Combined with GPS/IMU**	**Main characters**	**Application**	**Result model**
MIT city scanning (Teller et al., 2003)	Rectified images+ camera data	Yes	Image mosaic, semi-automatic	Visualization	Simple model
Sato et al. (2003)	Stereo image sequences	No	Multiple stereo, voxel voting	Scene recovery	Voxel
Stanford CityBlock (Roman et al., 2004)	Image sequences	No	Image mosaic from multiple viewpoints	Visualization	Multi-perspective images
Lensphoto (Zhang et al., 2007)	Multiple image sequences	Yes	Multi-baseline photogrammetry	Survey	Dense points
Cornelis et al. (2008)	Stereo image sequences	Yes	Surfaces from lines that are parallel to the gravity vector	Navigation	Wall plane, texture
Pollefeys et al. (2008)	Eight image sequences	Yes	GPU-based feature tracking, plane sweeping	Scene recovery	TIN, texture
Photo Tourism (Snavely et al., 2008)	Intersect images	No	SIFT points matching between image pairs, SFM	visualization	Dense points, texture
Xiao et al. (2009)	Multiple, high-overlap images	Yes	Semi-dense SFM, Multi-view semantic segmentation	Facade modeling	Rectangular building boxes

2.5 Texture mapping

There are a huge variety of buildings, while reconstructed models are always restricted to a limited level of detail and do not have realistic texture attributes of the building facades (Tsai et al., 2006). Generally, building reconstruction results are visualized in a wire-frame form or blocks, and with or without static orthophotos or artificial textures. Texture mapping can make the model appearing more realistic and compensate for lack of detail. Figure 2.6 shows an example of the same buildings with and without textures added.

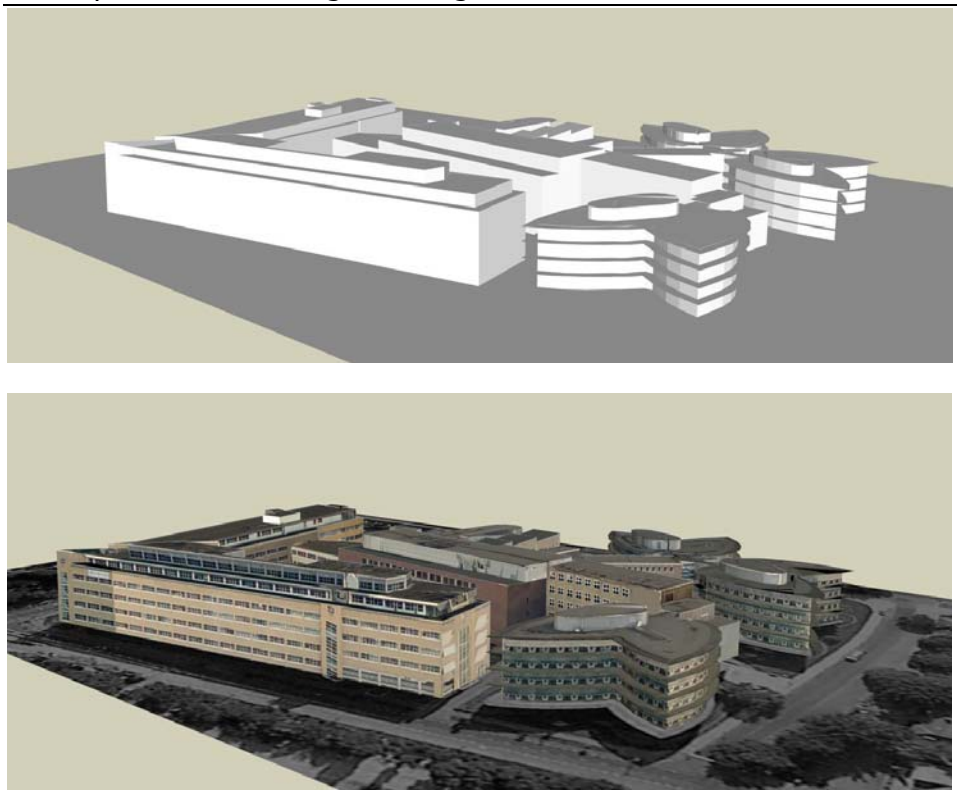


Figure 2.6 Building model with and without textures added

If the relationships between texture images and models have already been established, the first step in texture mapping is to select the best images. In (Fitzgibbon and Zisserman, 1998), the planes are textured by selecting the image from the sequence whose viewing direction is almost opposite to the normal vector of that plane. As the parameters of the interior and exterior orientation of images are known, images can be warped to remove any projective distortion. By projecting 3D surfaces to images, the corresponding image coordinates for polygon vertexes can be obtained. Then the 3D model could be textured by color values within the projected polygon. Although this seems a straightforward step, there are still many problems to tackle, such as radiometric image distortion, geometric scene distortion, and object occlusions (Remondino and El-Hakim, 2006).

In 3D textured models, radiometric distortions are presented along the edges of adjacent polygons textured using different images. These polygons can be model polygons or made by image mosaic. Histogram matching or equalization is popular in adjusting the color distributions of images into the same range (Du et al., 2001). Blending methods based on weighted functions can be used to achieve a smooth and seamless result (Debevec et al., 1996), such as alpha blending (Uyttendaele et al., 2001), pyramid blending (Adelson et al., 1984) or gradient domain stitching (Levin et al., 2004). Geometric scene distortion is generated from an incorrect reconstructed model or camera calibration, which can be avoided by improved reconstruction results and

system calibration. Some areas do not have good textures available in the provided data, which may be occluded either by other buildings, or by trees, cars or other objects. Such non-interested regions should be identified and removed from the texture mosaics and mended with the correct or similar texture blocks.

Occlusion areas can be identified from images by operators (Hays and Efros, 2007; Amirshahi et al., 2008). They can also be detected by comparing well registered images (Herley, 2005) or with dense surface data, such as laser scan points (Früh et al., 2005). Since, at times, trees can largely occlude the buildings behind them, special characteristics need to be considered. Greenness Index is widely used to separate areas blocked by green vegetation, as leaves are usually green or almost green (Tsai et al., 2006). By considering that most straight lines on facades are parallel to an axis in world coordinates, Kang (2007) made use of the occlusion by trees working against extracting lines from buildings.

To mend those removed areas is an image completion problem, which can be done though many images (Hays and Efros, 2007; Amirshahi et al., 2008) or by finding a similar area in the same image to replace it (Wilczkowiak et al., 2005a). Repeated patterns (e.g. windows) on a building facade are considered. It is possible to identify the areas of repeated patterns and their mirroring axis, so that the removed regions can be refilled by mirroring correctly textured blocks (Tsai et al., 2006). Jain (2002) provided a copy-paste method based on the idea of texture synthesis. The target area is presented as a hole. The method takes a window around the hole, finds a matching region in the image, and fills the hole by copying the matched region and pasting it over the hole.

When dealing with video data, the above problem can be identified as the problem of foreground-background segmentation, or background initialization, which is defined as follows: given a video sequence taken with a stationary camera, in which a static background is occluded by any number of moving objects in the foreground, output a single image of the static background (Colombari et al., 2007). In the simplest cases, foreground objects occlude background pixels for less than 50% of the entire sequence length. The median of each pixel color could be considered as a background value. The mixture of Gaussian (MOG) is used to model the pixel color distribution of the background. It can also be used with other algorithms, such as Bayesian frameworks (Lee et al., 2003), dense depth data (Harville, 2002), color and gradient information (Javed et al., 2002), and mean-shift analysis (Porikli and Tuzel, 2003). These methods may fail when the color distributions of the foreground and background are similar, or when the foreground is stationary for a long period of time.

Motion-based approaches have also been used. For instance, Wixson (2000) and Gutchess et al. (2001) proposed algorithms to detect salient motion by integrating frame-to-frame optical flow over time. These approaches assume that the object tends to move in a consistent direction over time, and that foreground motion has a different saliency. They may fail when there is no obvious motion difference between the foreground and background.

Other researchers (Toyama et al., 1999; Cristani et al., 2002; Harville, 2002) used region-based approaches by segmenting an image into regions or by refining low-level

classification obtained at the pixel level. An algorithm in (Zhong and Sclaroff, 2003) aimed to segment the foreground objects into dynamic textured backgrounds (e.g. water, escalators, waving trees, etc.). A real-time algorithm using a codebook model was presented in (Kim et al., 2005); it can detect moving objects from multiple backgrounds. However, for detecting highly occluded areas, pre-processing or semi-automatically selecting textures from the optimized ones are still more reliable than fully automatic methods.

2.6 Conclusion

From the literature review, it can be concluded that automatic building reconstruction, especially geometrical model reconstruction, is still the subject of ongoing research. The challenges are due to two aspects: inaccurate 3D interpretation of 2D images and the complexity of building structures.

Automatic image-based modeling methods rely on automatic feature extraction. Normally, feature detectors lose corners or miss edges, there is always some mismatching, and only sparse features can be extracted without dense matching. Therefore, reconstruction methods have to avoid error affection and correctly connect features. Uncalibrated and non-metric cameras are often used to capture terrestrial images, which are flexible to use in the ground-based environment. However, the lower image quality, in contrast to metric cameras, affects feature extraction. In terrestrial images, in particular, the robustness of feature matching and feature tracking is reduced by occlusions, illumination changes, limited locations of image acquisition, and un-textured surfaces.

Another challenge for automatic building reconstruction is that the buildings are dense and vary greatly in urban environments. Even from airborne images, recovering a complete, detailed, accurate and realistic 3D model is still a difficult task, especially when buildings have complex roof structures (Suveg and Vosselman, 2004; Remondino and El-Hakim, 2006). Compared with roofs in nadir airborne images, facades in terrestrial images contain more details, which leads to facade reconstruction from terrestrial images being more complex than roof reconstruction from airborne images. The basic requirement for building reconstruction methods is to ensure the geometric and topological correctness of the reconstructed building models, because 3D models with geometric and topological information are useful for further interpretation and applications. Existing automatic reconstruction algorithms lack a robust and comprehensive structure recovery.

Multiple images can provide more data to be used for validation and the reconstructed models may therefore be more accurate. Economic and flexible data acquisition procedures, together with the automatic structure from motion approach, are the main advantages of using video image sequences as the source data for building reconstruction. The literature review shows that recovering structure from motion techniques has already established mathematic theories and general solutions. Thus, methods of recovering the accurate structure of buildings in a realistic environment should be given much attention. Such knowledge can be widely used in many image

analysis methods and it may be used to describe any kind of information (Baltsavias, 2004). Since the majority of buildings satisfy the assumption that they can be modeled geometrically as an ensemble of planar polygonal surface patches, integrating the knowledge of building structure into the reconstruction could be a useful solution. Furthermore, polyhedral models seem to be a relatively simple and efficient way of presenting building structures.

3 Methodology and approach

3.1 Outline

Video image sequences can show many building details, but it is only in two dimensions, and 3D information may not be correctly extracted due to wrong image interpretation. The literature review (Chapter 2) showed that to take advantage of the flexibility of a data-driven approach and the robustness of a model-driven approach would be a good alternative for recovering building structures. In the aims mentioned for this PhD research, the emphasis was on how to recover the shape of buildings with planar structures, and not specific architectures. In order to cope with the complexity of real scenes, a hybrid data-driven/model-driven method is proposed, which takes surfaces that are grouped from extracted feature points and edges as the basic elements for model reconstruction. This enables the method to recover building models in a generalized way, but it is not limited to predefined solid types. It also exploits geometric and topological constraints to overcome the problems caused by occlusions, low contrast, noise and disturbances. There are two main issues: one is how to make use of edges to add shape constraints to surfaces and increase the accuracy of surface; the other is how to define and use the knowledge of building structures to recover buildings from image sequences. To solve these issues, a new method is proposed and an overview of the method is given in this chapter.

3.2 Related concepts

In this section, some of the concepts related to the proposed building models and the prior knowledge which is used in the developed strategy are specified.

Surface patch: A surface patch has a closed outline, represented by a polygon. The normal vector of the surface patch points to the outside of the building. For example, there is a building shown in Fig. 3.1 and six surface patches ($s_i, i = 1 \cdots 6$) of it are visible from the viewing point. The normal vectors of these surface patches must be in the direction indicated in the drawing and the sequences of vertexes are consistent with the normal vectors in a right-hand coordinate system. Each surface patch (s) has a number of attributes, such as color, or material. In the context of this thesis, except for semantic meaning, only geometric attributes as extracted from images are considered further. In particular, these are position, orientation and shape.

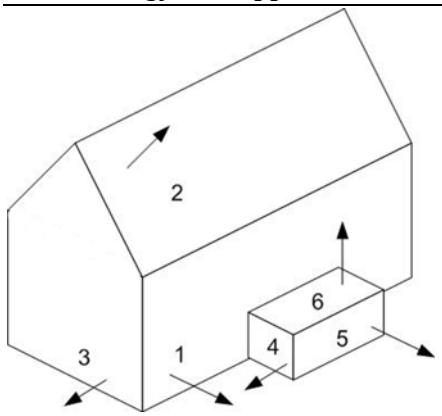


Figure 3.1 Surface patches with their normal vectors

Surface patch neighboring relation: There are two kinds of neighboring relations between surface patches. One is that two surface patches can be part of a closed entity and their normal vector points to the outside of the entity, such as surface (1, 2), (1, 3), (2, 3), (4, 5), (4, 6) and (5, 6) in Fig. 3.1. The other relation type is the edge of one surface is contained in another surface and they belong to two different closed entities, e.g. surface 4 and 6 are attached to surface 1. To identify neighboring relations, spatial relations between surface patches need to be defined (cf. section 3.3).

Local model: A local model is constructed by three adjacent surface patches, and defines a part of a volume. Each local model has exactly one corner point where the three planes intersect, i.e. when representing the model as a graph, the corner is a node with degree three (Fig. 3.2).

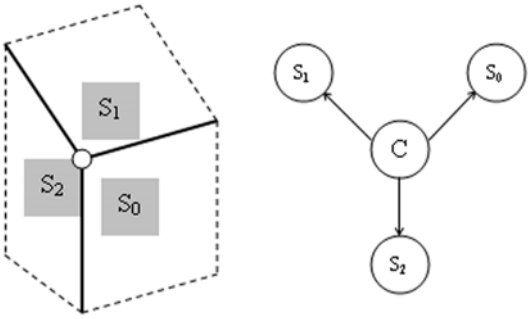


Figure 3.2 Local model showing the topological relation of three surface patches (left). Drawn as a graph (right), the arrows express the neighborhood relations.

Knowledge: Rules and constraints are retrieved from a generic building facade structure. In this thesis, the target objects are limited to some standard buildings, thus those buildings that do not comply with the following specifications will not be considered further:

- All building faces are planar.
- Walls are vertical.

- Roofs intersect with walls.

From the structural information of buildings, some *preferential* knowledge, which can be used to enhance the modeling, is predicted as follows.

- Building ground planes mostly have rectangular corners.
- Buildings exhibit a high degree of self-similarity and regularity. To define this preferential knowledge is motivated by observations that buildings are often designed having economical, manufacturing, functional, or aesthetic considerations in mind.

The preferential knowledge provides essential guidance during fitting of the surface patch outline, finding the adjacent surface patches, and also for formulating the local model hypotheses.

3.3 Knowledge presentation

Two-dimensional topological relationships have been intensively studied by Egenhofer and Franzosa (1991). As the extension of 2D topology, 3D topological relationships are much more complex, because objects' relations in 3D space have a number of similar but different situations and 3D-2D, 2D-2D, even 3D-1D relations also need to be considered (Pigot, 1991; Pilouk, 1996). It is very difficult to determine a general classification to describe all situations (Ellul and Haklay, 2006). Some useful subdivisions according to applications for this research are therefore specified, based on relationships as proposed in Egenhofer and Herring (1990) and Pigot (1991).

Semantic information: The main surface patches belong to walls, roofs, ground planes or extrusions. Extrusions include balconies and overhanging parts on the wall or roof. So, $\forall s, Semantic(s) = Wall \vee Roof \vee Ground \vee Extrusion$.

Position: Some surface patches can be expected at relative positions inside a scene, e.g. the ground is usually the lowest part and other types might be close to the ground. A 3D boundary box of the area of interest can be calculated from point and edge features. Surfaces exclusively located in the lower half part of the buildings' vertical extension are considered to be "low".

$$\forall s, Position(s) = Low \vee NotLow$$

Edges (e) in the 2D box area can also be presented as:

$$\forall e, Position(e) = Low \vee NotLow$$

Orientation: The orientation of a semantic surface patch is predictable, e.g. the ground is horizontal, walls and extrusions are usually vertical, and roofs are not vertical. And they can be presented as:

$$\forall s, Orientation(s) = Horizontal \vee Vertical \vee Sloped$$

Similar presentations are used for edges in 2D:

$$\forall e, Orientation(e) = Horizontal \vee Vertical \vee Sloped$$

Shape: Building surface patches have a regular and common shape, especially when buildings are presented in a generalized form. Triangle, rectangular, parallelogram, and trapezoid are basic shapes for surface patches. Building surface patches may also be seen as a combination of these basic shapes. For the ground, a convex hull is chosen to present it.

$$\forall s, Shape(s) = Rectangular \vee Triangular \vee Trapezoid \vee Parallelogram \vee Others$$

Some spatial relations between surface patches can have a numerical value, such as the **angle** between two surface patches and the **distance** between two surface patches. Others attributes are:

Adjacent: When the distance between two surface patches is zero, they are adjacent.

$$\forall s_1, s_2, Adjacent(s_1, s_2) = \begin{cases} True, & if \quad distance(s_1, s_2) == 0 \\ False & else \end{cases}$$

Intersection: An intersection is a basic spatial relation between two surface patches, which is different from **through** (Pigot, 1991). Intersection surface patches are adjacent and some points are located on both of them. Two intersection types are specified according to the location of common edges on each surface (Fig. 3.3).

$$\forall s_1, s_2, \quad Intersection(s_1, s_2) = ShareBoundaryEdge \vee NotShareBoundary\,Edge$$

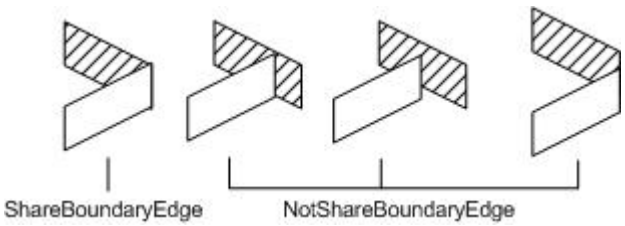


Figure 3.3 Examples of intersection relations between surface patches

Consistency: Consistency is the relation between normal vectors of two adjacent surface patches. Surface consistency means the normal vectors of surface patches should be consistent to a point interior or exterior to the volume. As shown in Figs. 3.4a and b, two adjacent surface patches are consistent along their common edge if:

$$[l, v_1, n_1] \cdot [l, v_2, n_2] < 0 \qquad (3.1)$$

where l represents the direction of the common edge,

v_i represents the orthogonal direction from the common edge to the interior of surface patch,

n_i represents the normal vector of the surface patch,

$[l, v_i, n_i]$ stands for the scalar triple product.

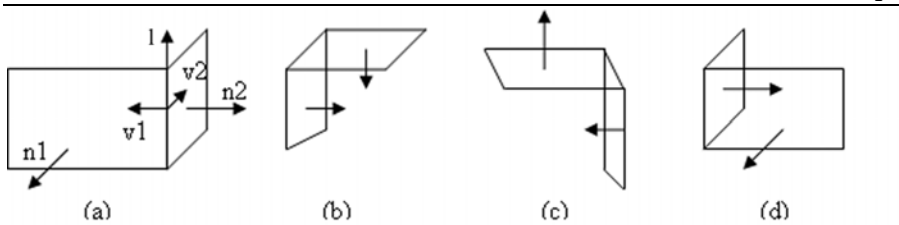


Figure 3.4 Coherent (a,b) and non-coherent (c,d) adjacent surfaces

Fig. 3.4 also shows some counter-examples in (c) and (d). So,

$$\forall s_1, s_2, Intersection(s_1, s_2) = ShareBoundaryEdge$$
$$\Rightarrow Consistency(s_1, s_2) = True \vee False$$

Finally, surface patch neighborhood relations can be defined as: $Relation(s_1, s_2) = Meet \vee Attach$

These neighborhood relations are subdivisions of the original **meet** relationship (Egenhofer and Herring, 1990) and can be derived from other attributes:

$$\exists s_1, s_2, Relation(s_1, s_2) = Meet$$
$$\Leftrightarrow Intersection(s_1, s_2) = ShareBoundaryEdge \wedge Consistency(s_1, s_2) = True$$
$$\exists s_1, s_2, Relation(s_1, s_2) = Attach$$
$$\Leftrightarrow Intersection(s_1, s_2) = NotShareBoundaryEdge \vee$$
$$(Intersection(s_1, s_2) = ShareBoundaryEdge \wedge Consistency(s_1, s_2) = False)$$

The preferential knowledge, which is used as guidance during building model reconstruction when some surface patches have not been grouped from features, can be presented as follows.

- Building ground planes mostly have rectangular corners. With this knowledge, if a wall and a roof can make a local model hypothesis and the other wall is invisible from available images, a new wall that is orthogonal to the extracted wall is estimated:

$$\exists s_1, Semantic(s_1) = Roof, \quad \exists s_2, Semantic(s_2) = Wall \quad \Rightarrow$$
$$if \quad Relation(s_1, s_2) = Meet \quad \wedge \quad \neg(\exists s, Relation(s, s_1) = Meet \wedge Relation(s, s_2) = Meet), \quad then$$
$$\{s\} = \{s\} + s_n \wedge Angle(s_n, s_2) = RightAngle \wedge Relation(s_n, s_1) = Meet \wedge Relation(s_n, s_2) = Meet$$

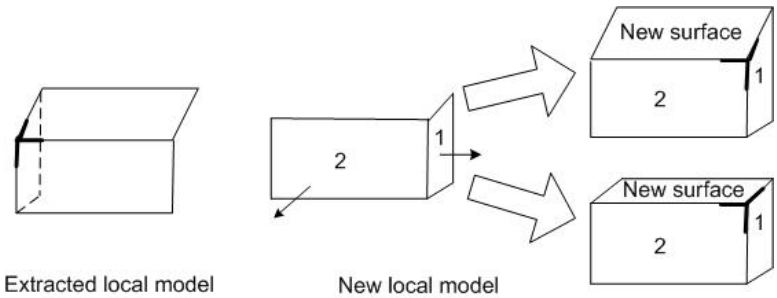


Figure 3.5 Example for making roof hypothesis under preferential knowledge

- If a sloping roof has been extracted and there are two walls at other parts of the buildings that can form a local model, then the roof that connects with these two walls can be sloping (Fig. 3.5). Since this new roof surface is recovered under hypothesis, its plane parameters need to be verified (see section 5.2 for more details).

$$\exists s_1, Semantic(s_1) = Roof, Orientation(s_1) = Sloped, \quad \exists s_2, s_3, Semantic(s_2) = Semantic(s_3) = Wall$$
$$\Rightarrow if \quad Relation(s_2, s_3) = Meet \quad \wedge \quad \neg(\exists s, Relation(s, s_2) = Meet \wedge Relation(s, s_3) = Meet), \quad then$$
$$\{s\} = \{s\} + s_n \wedge Relation(s_n, s_1) = Meet \wedge Relation(s_n, s_2) = Meet \wedge Orientation(s_n) = Sloped \vee Horizontal$$

3.4 Workflow

The workflow consists of the following steps (cf. Fig. 3.6):

1. Preprocessing. After feature tracking across the sequence, the projection matrices and 3D coordinates of feature points are computed through bundle adjustment, and the lens and image distortions are corrected for, i.e. undistorted images are used thereafter (see section 3.5).

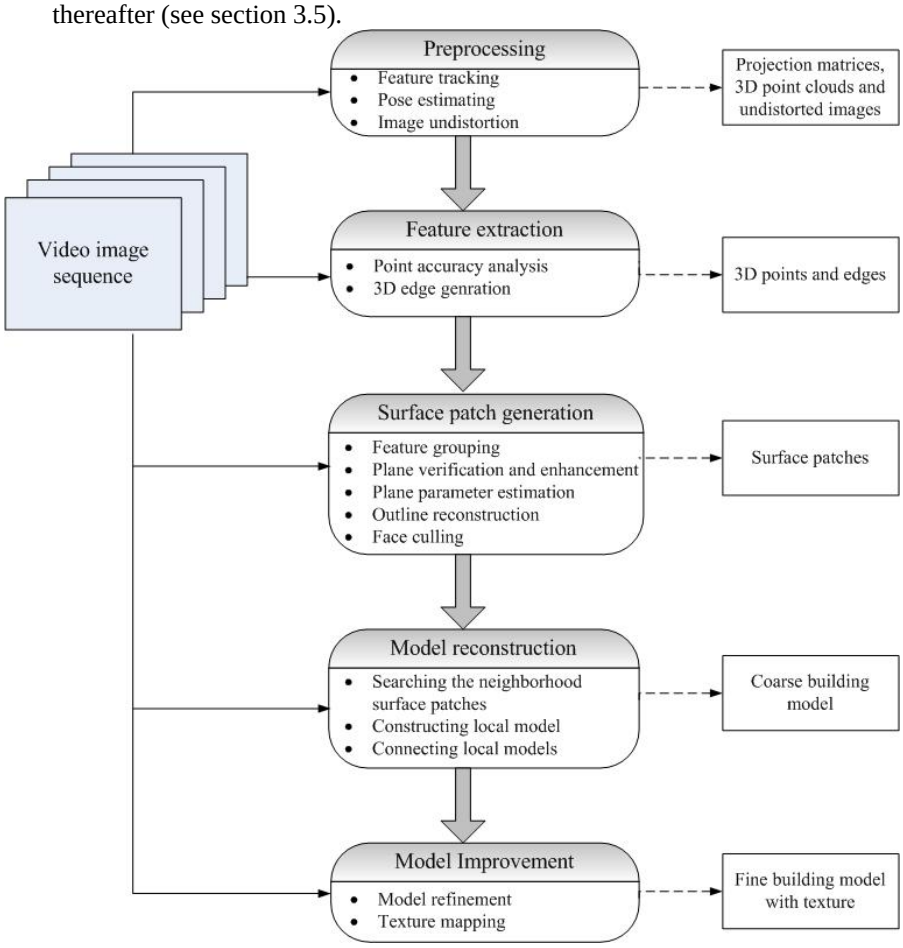


Figure 3.6 Modeling pipeline for building reconstruction from image sequence

2. Feature extraction. Starting from 3D points tracked from a video image sequence, the point accuracy is analyzed first to obtain reliably matched points (section 4.2). In order to introduce more constraints for the reconstruction and to fill the gaps in 3D point clouds, 3D edges are also used as primitives for the reconstruction. Only edges near the reliably matched points are considered further, because these points can provide geometric constraints for searching for corresponding edges. More information can be found in section 4.3.

3. Surface patch generation. Section 4.4 describes how surface patches are recovered from extracted geometric features. Firstly, extracted 3D points and edges are grouped and verified according to predefined rules. After estimating plane parameters from all the edges and points in the plane, the knowledge on the generic shapes of building surfaces guides the outline generation. Finally, the normal direction is defined to point outside of the building.

4. Coarse model reconstruction. For connecting surface patches, building structure knowledge is integrated into the model reconstruction. Adjacent surface patches are searched first (section 5.3). Then local models are recovered by coherent adjacent surface patches (section 5.4), and finally, local models are connected to form a complete model (section 5.5). The topological relation between surface patches is set up during construction of the local model and helps in connecting different local models.

5. Model improvement. In the above building reconstruction method, some assumptions about structures are made that are not always fully satisfied by some buildings. Section 5.6 describes the method to refine coarse polyhedral models computed in the previous sections by attempting to fit detailed hypotheses to features that are not fixed by the coarse model. The quality of coarse building models can therefore be improved by checking image intensity in the local area (section 5.6). Finally, exposed (non-occluded) facade textures are recovered from the image sequence. Textured models can thus provide a realistic visualization of the buildings (section 5.7).

3.5 Preprocessing

In most cases, when applying the method described in this thesis, uncalibrated, non-metric cameras are used and precise navigation information through GPS/IMU is not available. Thus, there is no further information about the image exterior or interior orientation. The initial step consists in relating the images to each other. Usually, when dealing with video image sequences, this step is done through feature tracking and then via camera calibration and orientation. Lens distortion can have a serious effect on the quality of camera calibration and orientation and, if left uncorrected, can make it impossible to get a usable camera orientation.

Figure 3.7 The original image with distortion (resolution 720×576)

In the ideal, pinhole camera model, straight edges are in reality mapped as straight edges in the images. As shown in Fig. 3.7, distortion makes objects appear to change shape, which results in straight edges being mapped as curved edges. The lens distortion includes both radial distortion and tangential distortion. For commercial, non-metric cameras, only radial distortion needs to be considered (Feng, 2002) and it can be corrected using Brown's distortion model (Brown, 1966).

$$\begin{aligned} x_u &= x_d + (x_d - x_c)(k_1 r^2 + k_2 r^4 + \cdots) \\ y_u &= y_d + (y_d - y_c)(k_1 r^2 + k_2 r^4 + \cdots) \end{aligned} \tag{3.2}$$

where k_n is the nth radial distortion coefficient,

(x_d, y_d) is the pixel position in the original image,

(x_u, y_u) is the new pixel position when lens distortion is removed,

(x_c, y_c) is the principal point position in pixels,

$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$.

Since most radial distortion affection can be removed by value k_1, a single-parameter model is chosen.

$$\begin{aligned} (x_d - x_c)(1 + kr^2) &= (x_u - x_c) \\ (y_d - y_c)(1 + kr^2) &= (y_u - y_c) \end{aligned} \tag{3.3}$$

Figure 3.8 The undistorted image (resolution 784×628)

The distortion can be corrected by warping the image with a reverse distortion. This process results in a new set of images that are of a different size to the originals. They are larger if the value of k is positive and smaller if the value is negative. An example is shown in Fig. 3.8. For the implementation of the workflow at hand, the commercial software Boujou (Vicon, 2011) is currently being used. Long, straight line features in buildings can be used to manually estimate the distortion. After adding a calibration line, the value k is adjusted to remove the image's distortion, until the feature that should be straight is parallel to the calibration line. See (Dobbert, 2005) for detailed information on the approach as implemented in Boujou.

For the feature tracking step, Boujou applies the KLT tracker (Lucas and Kanade, 1981), which is the one most widely used, to determine 2D-2D point correspondences in consecutive video frames. Further information on structure from motion can be found in section 2.2 or from some references, such as (Hartley and Zisserman, 2004; Nistér, 2004b). Besides performing the fully automatic reconstruction up to scale, in Boujou it is possible to define a coordinate frame and constraints on the actual scene geometry, like known distances in object space between feature points. As the result of preprocessing, camera information such as interior orientation parameters (assumed to be constant throughout the sequence), external camera orientations, and 3D point clouds are calculated. The corresponding projection matrix for each frame and undistorted images are also available.

4 Feature extraction and grouping

Feature extraction is used to extract important image information (e.g. points, edges and regions), which has locally distinguishable chromatic properties and results in 3D features. This low-level processing, which determines the primitives for the further steps, is as important as the high-level processing (reconstruction) during the modeling of buildings.

4.1 Feature selection and presentation

Geometric features usually include points, edges and regions. In this thesis, points and edges are considered to be basic features that can be extracted from image sequences and used to find accurate correspondences. Region detection is usually the result of image segmentation, which detects the image areas that fulfill a certain similarity criterion, such as the similar intensity value. As the projected shape of a region varies depending on perspective transformation, and it is difficult to build up geometric correspondences of regions in images, regions are not accurate or simple enough to be used as the basic primitives. Points and edges are considered to be basic features since the point clouds obtained from applying feature tracking and camera orientation steps to a video sequence are not dense enough to allow a complete description of a 3D scene and not all the boundary vertices for object reconstruction, such as corner points, can present.

For easily dealing with mathematic computation and implementation, a feature is described in different forms when it is necessary in this thesis. How to convert features from different forms is also defined in this section. A 2D point $\boldsymbol{x} = (x,y)^T$ on the Euclidean plane can be presented in a homogeneous form: $\boldsymbol{x} = (x,y,1)^T$. For a 2D line, the homogeneous form is: $ax + by + c = 0$, then the parameters can be written in a homogeneous vector $\boldsymbol{l} = (a,b,c)^T$. An angle-distance form describes the 2D line in a similar way: $\boldsymbol{l} = (cos(\theta), sin(\theta), -d)^T$. A 2D line can also be defined by a foot point ($\boldsymbol{x}_f$) and a direction vector ($\boldsymbol{n}$), and each point can be represented by the corresponding scalar (t).

$$\boldsymbol{l} = \boldsymbol{x}_f + \boldsymbol{n} * t \tag{4.1}$$

Therefore, a 2D edge can be described in this form with two endpoint scalars.

Points in 3D are represented analogously to points in 2D: given a 3D point $\boldsymbol{X} = (X,Y,Z)^T$, the homogeneous representation is $\boldsymbol{X} = (X,Y,Z,1)^T$. Assume a 3D edge

$\boldsymbol{L}$ which is formed by two endpoints $\boldsymbol{X}_0, \boldsymbol{Y}_0$ (Fig. 4.1a), it can be represented in a similar way as a 2D edge in equation 4.1 (Fig. 4.1b).

$$\boldsymbol{L} = \boldsymbol{X}_f + \boldsymbol{N} * t, \quad t_{min} \le t \le t_{max} \tag{4.2}$$

where t_{min}, t_{max} are the scalars of two endpoints.

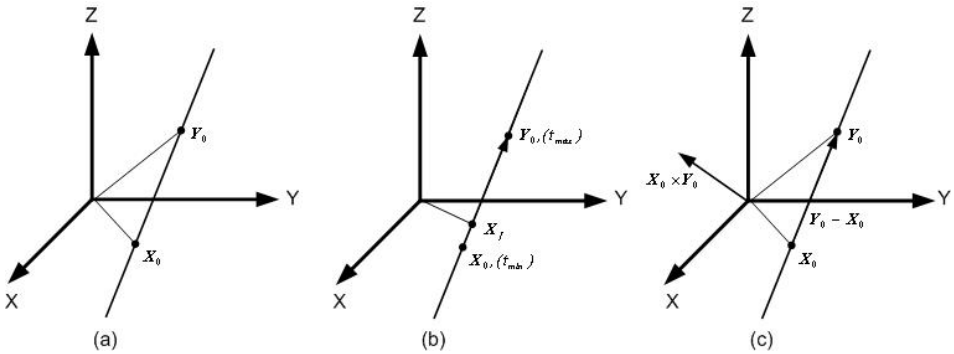


Figure 4.1 Three different presentations for a 3D edge

There exist a number of other representations for 3D lines, such as the one shown in Fig. 4.1(c), a 6-vector $\boldsymbol{L} = (L_1, L_2, L_3, L_4, L_5, L_6)^T$ in Plücker coordinates. The homogeneous part $\boldsymbol{L}_h = (L_1, L_2, L_3)^T$ is constrained to be orthogonal to the Euclidean part $\boldsymbol{L}_o = (L_4, L_5, L_6)^T$, i.e. $\boldsymbol{L}_h^T \boldsymbol{L}_o = 0$. This constraint is called the Plücker constraint of a 3D line, see (Hartley and Zisserman, 2004; Heuel, 2004)) for more details. The homogeneous part represents the line direction and the Euclidean part decides the distance from the origin to the line. Thus the 6-vector $\boldsymbol{L}$ has 4 degrees of freedom, given both the orthogonal and homogeneous constraints.

$$\sqrt{L_1^2 + L_2^2 + L_3^2} = 1 \tag{4.3}$$

$$L_1 L_4 + L_2 L_5 + L_3 L_6 = 0 \tag{4.4}$$

Assuming $\boldsymbol{X}_o(X_1, X_2, X_3)$ is the projection point of origin on a 3D line and $\boldsymbol{Y}_0(Y_1, Y_2, Y_3)$ is the point with scalar 1, if a line in Plücker coordinates is given, the relations between the parameters can be represented as:

$$\begin{aligned} Y_1 &= L_1 + X_1 \\ Y_2 &= L_2 + X_2 \\ Y_3 &= L_3 + X_3 \end{aligned} \tag{4.5}$$

and

$$\begin{aligned} L_4 &= X_2 Y_3 - X_3 Y_2 \\ L_5 &= X_3 Y_1 - X_1 Y_3 \\ L_6 &= X_1 Y_2 - X_2 Y_1 \end{aligned} \tag{4.6}$$

Combining equations 4.4, 4.5 and 4.6 together, point $\boldsymbol{X}_o$ can be calculated:

$$X_1 = \frac{L_2L_6 - L_3L_5}{L_1^2 + L_2^2 + L_3^2}$$
$$X_2 = \frac{L_3L_4 - L_1L_6}{L_1^2 + L_2^2 + L_3^2} \qquad (4.7)$$
$$X_3 = \frac{L_1L_5 - L_2L_4}{L_1^2 + L_2^2 + L_3^2}$$

Therefore, from one point $\boldsymbol{X}_0$ on the line with the direction $(L_1, L_2, L_3)^T$, the line can be determined in 3D space coordinates. Converting a 3D line from 3D space to the Plücker coordinates can be done by taking any two points on the line.

4.2 Point accuracy analysis

During the preprocessing steps, camera projection matrices for each frame are obtained with some corresponding points in 2D and 3D. As described in section 3.5, the method is based on the most common model for cameras, i.e. the pinhole camera model. A point in 3D space is projected into an image by computing a viewing ray from the unique projection center to the point and intersecting this viewing ray with a unique image plane. In the case of a perfect camera calibration and a perfect matching of the corresponding points in the images, all the rays could intersect in a common point, i.e. the object point (Fig. 4.2). For uncalibrated image sequences, it is not usually the case that the rays of the corresponding points in the images intersect precisely in a common point. The points' quality should therefore be analyzed first to remove any false points that do not coincide with real objects.

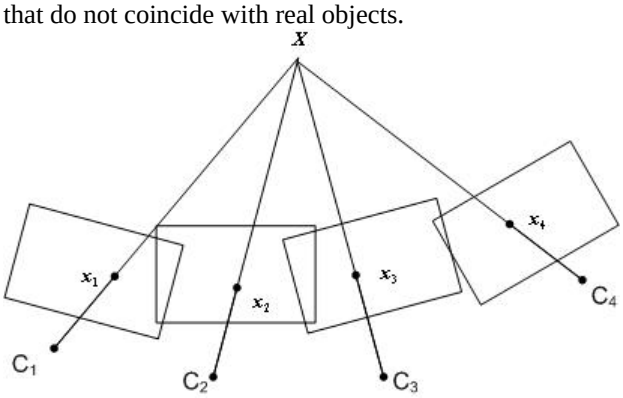


Figure 4.2 Multiple camera views

As a primary accuracy assessment, visual cues were evaluated; examples of visual cues include (Dobbert, 2005):

- 3D points should move with the object that they represent throughout the image sequence.

- When viewed in 3D, the camera perspective should appear correct relative to the objects in the scene and the virtual camera path should follow the height and alignment of the original.

Error can be evaluated quantitatively by analyzing residual error values or by analyzing projection error. Projection error is the difference between the position of the original feature point and the 3D points as they are projected using the estimated camera parameters. Assuming a 3D point $\boldsymbol{X}(X,Y,Z)$ is visible in $n+1$ images, a set of corresponding image points $\boldsymbol{x}_i(x_i,y_i)$, $i=0,\cdots,n$, and camera projection matrices $\boldsymbol{P}_i$, $i=0,\cdots,n$, for each frame in which the 3D point is visible are known. As the relation between the 3D point and its corresponding image point is

$$\boldsymbol{x}=\boldsymbol{PX} \tag{4.8}$$

by projecting a 3D point from object space to the image plane i, the calculated image point $\boldsymbol{x}'_i(x'_i,y'_i)$, $i=0,\cdots,n$, can be obtained. The difference d between the tracked image points (x,y) and corresponding calculated image points (x',y') can be expressed by

$$d=\sqrt{(x-x')^2+(y-y')^2} \tag{4.9}$$

Then the standard deviation σ of image points corresponding to point (X,Y,Z) can be calculated by

$$\sigma^2=\frac{\sum_{i=0}^{n}d_i^2}{n},i=0,\cdots,n \tag{4.10}$$

With $\boldsymbol{P}=\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix}$, equation (4.8) can be written as:

$$\begin{aligned} x&=\frac{p_{11}X+p_{12}Y+p_{13}Z+p_{14}}{p_{31}X+p_{32}Y+p_{33}Z+p_{34}} \\ y&=\frac{p_{21}X+p_{22}Y+p_{23}Z+p_{24}}{p_{31}X+p_{32}Y+p_{33}Z+p_{34}} \end{aligned} \tag{4.11}$$

The 3D coordinate of point (X,Y,Z) is estimated by intersecting all its viewing rays as

$$\begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{pmatrix}=\boldsymbol{A}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{4.12}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \left(\frac{\partial x}{\partial X}\right)_1 & \left(\frac{\partial x}{\partial Y}\right)_1 & \left(\frac{\partial x}{\partial Z}\right)_1 \\ \left(\frac{\partial y}{\partial X}\right)_1 & \left(\frac{\partial y}{\partial Y}\right)_1 & \left(\frac{\partial y}{\partial Z}\right)_1 \\ \vdots & \vdots & \vdots \\ \left(\frac{\partial x}{\partial X}\right)_n & \left(\frac{\partial x}{\partial Y}\right)_n & \left(\frac{\partial x}{\partial Z}\right)_n \\ \left(\frac{\partial y}{\partial X}\right)_n & \left(\frac{\partial y}{\partial Y}\right)_n & \left(\frac{\partial y}{\partial Z}\right)_n \end{bmatrix} \tag{4.13}$$

The partial derivatives are computed from equation 4.11 and the covariance matrix C for 3D point can be obtained by

$$\boldsymbol{C} = \sigma^2 \cdot \left(A^T A\right)^{-1} \tag{4.14}$$

Then, the theoretical precision of the computed 3D point can be expressed as error σ_{3D} according to

$$\sigma_{3D}^2 = \sum_{i=0}^{2} C_{ii} \tag{4.15}$$

If σ_{3D} is larger than a suitable threshold, the 3D point is not accurate. Figure 4.3 shows the reliable (black) and less reliable points (red) resulting from the precision analysis.

Figure 4.3 Point analysis result: reliable points are shown in black, less reliable points in red

4.3 Edge generation

Existing approaches for edge detection can obtain acceptable results. However, edge matching is not always accurate for several reasons. The first reason is the deficiencies in extracting edges and their connectivity: although the orientation of an edge can be recovered accurately, the endpoints are not reliable, and furthermore the topological connections between edges are often lost during segmentation. The second reason is that there is no strong, disambiguous and simple geometric constraint available over more than three views. In the research performed for this thesis, reliably matched points are used as guidance for edge matching, which means only edges near to these points are considered.

4.3.1 Edge extraction

Edges are first detected in each frame separately. To start, an 8-bit binary edge map is generated in each image by running an edge detector, such as the Canny detector or EDISON edge detector (Canny, 1986; Meer and Georgescu, 2001). Refer to section 2.1.2.1 for more details and parameter settings. Figures 4.4 and 4.5 show the edge detection results with some appropriate parameters. The edge detectors were applied to the same image with a resolution of 2448×1836. There is no significant difference between these two results. For image sequences, most of the time, the Canny detector is therefore selected due to its simpler parameter setting.

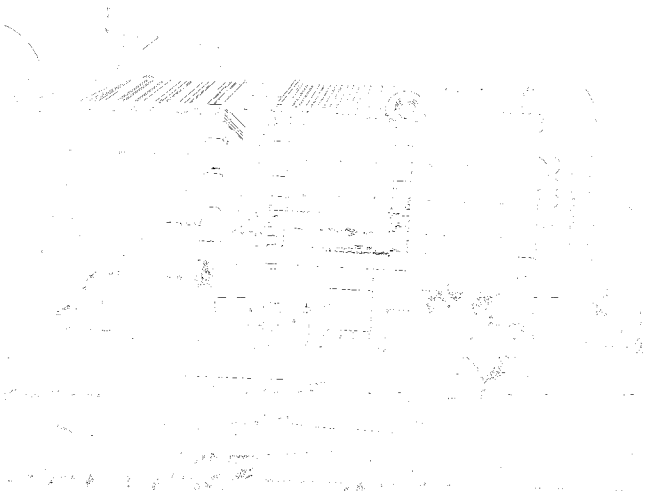


Figure 4.4 The Canny edge detection result

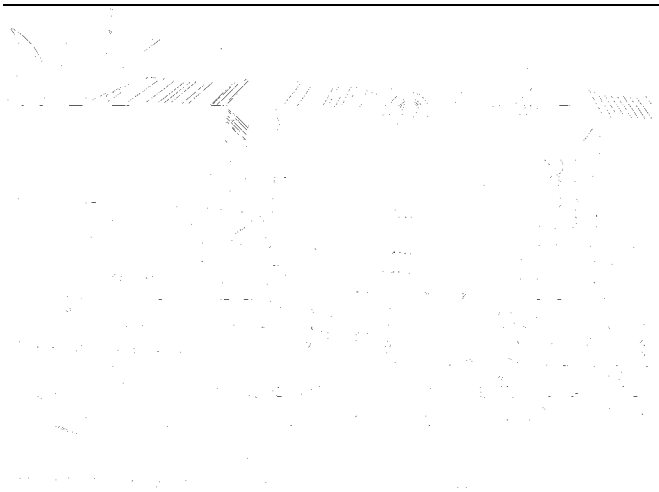


Figure 4.5 The EDISON edge detection result

The second step is to use a Hough transform to extract straight edges from the edge map. Projecting straight edges to the original image, the edge extraction result can be shown as in Fig. 4.6.

Figure 4.6 The edge extraction result

Since different parameters lead to quite different edge detection and straight line extraction results, it is difficult to get a perfect result so that all the edges are extracted correctly, especially the position of any endpoints. So, in this step, a result that could include most of the obvious straight edges on the building facade is considered an appropriate result.

4.3.2 Edge matching

After edges have been extracted from each frame, the relationship between these edges and reliably matched points will be obtained by distance analysis. Using reliably matched points to guide the edge matching is the key point in this method, only edges near these good quality points are considered, which reduces the search space for corresponding 2D edges in frames. The workflow is described below and shown in Fig. 4.7.

1. Project a reliable 3D point to an image in which it is visible (or using a corresponding image point), and calculate the distance between this point and edges detected in the same image. The distance here is the distance between a point and a finite edge. If the distance is less than a threshold value (according to image quality and feature extraction result, e.g. one pixel), the edge is considered as an edge candidate in that image.

2. Use the same method described in step 1 to analyze edges from all the images in which the same 3D point is visible. With this measurement, edge candidates in images are obtained. This method is much faster than applying an epipolar beam from the endpoints to find candidates.

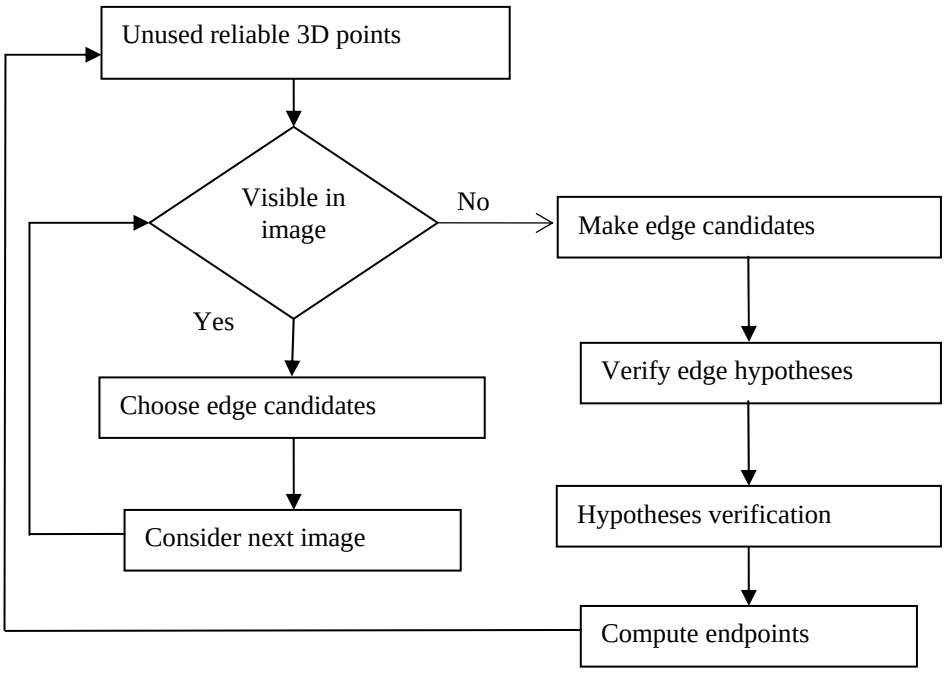


Figure 4.7 Outline of edge matching

3. In order to enlarge the baseline to get a more accurate result, a 3D edge hypothesis is made between candidate edges from the first and last images. Since the corresponding edges usually cannot be extracted in every image, and also considering the computation time, the candidate edges from the first and last 10%

of images are chosen. A 3D infinite edge hypothesis is the intersection of two planes $\boldsymbol{A}$ and $\boldsymbol{B}$, each defined by one optical center and the corresponding 2D edge.

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{P}_a^T \boldsymbol{l}_a \\ \boldsymbol{B} &= \boldsymbol{P}_b^T \boldsymbol{l}_b \\ \boldsymbol{L} &= \boldsymbol{A} \cap \boldsymbol{B} \end{aligned} \tag{4.16}$$

where $\boldsymbol{l}_a, \boldsymbol{l}_b$ are 2D edges in image a and image b,

$\boldsymbol{P}_a, \boldsymbol{P}_b$ are projection matrixes of image a and image b,

$\boldsymbol{L}$ is the intersection of plane A and plane B.

4. Project the 3D infinite edge to each image. As a projection matrix P for points is known, $\boldsymbol{x} = \boldsymbol{PX}$, it is able to construct a projection matrix Q that can be applied to 3D edges, $\boldsymbol{l} = \boldsymbol{QL}$, where $\boldsymbol{Q}$ is a 3×6 matrix. The row of projection matrix $\boldsymbol{P}$ can be interpreted as three distinct planes A, B, and C, intersecting in the projection center. Therefore, the projective camera matrix for points can be written as:

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{A}^T \\ \boldsymbol{B}^T \\ \boldsymbol{C}^T \end{pmatrix} \tag{4.17}$$

The projective camera matrix for lines is given as:

$$\boldsymbol{Q} = \begin{pmatrix} \overline{(\boldsymbol{B} \cap \boldsymbol{C})}^T \\ \overline{(\boldsymbol{C} \cap \boldsymbol{A})}^T \\ \overline{(\boldsymbol{A} \cap \boldsymbol{B})}^T \end{pmatrix} \tag{4.18}$$

where $\overline{\boldsymbol{B} \cap \boldsymbol{C}}$ is the dual intersection of the two planes B and C. More details are given in (Hartley and Zisserman, 2004; Heuel, 2004).

Calculate distance and angle between the projection results and edge candidates. If the distance and the angle is less than a predefined threshold, the edge candidate is considered as a corresponding edge for the 3D edge hypothesis.

Compare the number of corresponding edges with the number of images considered. Since an object edge cannot always be detected in every image, if the rate is higher than a threshold (50% by default), the hypothesis is confirmed. Otherwise it should be rejected and a new hypothesis needs to be made from the edge candidates. Return to step 3.

5. When the hypothesis is confirmed, the corresponding edge in each image can be retrieved. From these 2D edges, 3D edge estimation is done, see section 4.3.2.1 below. The 3D edge can still be rejected if the estimated variance factor is larger than a suitable threshold or if the solution does not converge.

6. Compute endpoints for the estimated 3D edge. By backwardly projecting rays from the endpoints of the corresponding 2D edges and taking the intersection with the

estimated 3D edge, two sets of endpoint candidates for the 3D edge can be obtained. The method described in section 4.3.2.2 is used to fix the endpoints.

7. Take the next reliable 3D point, until all the points have been processed.

4.3.2.1 3D Edge Estimation

The geometric construction can be described as an estimation task, where an unknown 3D edge has to be fitted to a set of 2D edges from different images. So a relation between a 3D line and 2D line can be defined as:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = Q \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \end{pmatrix} \tag{4.19}$$

The relation to the angle-distance form of a 2D line is given by a multiplication factor $1/\sqrt{a^2+b^2}$:

$$l = \begin{pmatrix} cos(\theta) \\ sin(\theta) \\ -d \end{pmatrix} = 1/\sqrt{a^2+b^2} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{4.20}$$

If there are n lines matched across the image sequence, the 3D edge can be estimated by using a Gauss-Markoff model with constraints: N=3n observations l for U=6 unknown parameters L in Plücker coordinates with H=2 constraints h.

$$l + \hat{v} = f\left(\hat{L}\right) \tag{4.21}$$

$$h(\hat{L}) = 0 \tag{4.22}$$

In order to obtain the corrections Δl and ΔL , the following Jacobians are needed:

$$A = \left.\frac{\partial f(L)}{\partial L}\right|_{L = L^0} \tag{4.23}$$

$$H = \left.\left(\frac{\partial h(L)}{\partial L}\right)^T\right|_{L = L^0} \tag{4.24}$$

An initial covariance matrix C_{ll} of the observed 2D edges can be calculated from the uncertainty of edge extraction result. More details are given in (Heuel, 2004). So,

$$\begin{bmatrix} A^T C_{ll}^{-1} A & H \\ H^T & 0 \end{bmatrix} \begin{bmatrix} \Delta\hat{L} \\ \mu \end{bmatrix} = \begin{bmatrix} A^T C_{ll}^{-1} \Delta l \\ c_h \end{bmatrix} \tag{4.25}$$

$$\hat{v} = -(\Delta l - A\Delta\hat{L}) \tag{4.26}$$

where $\Delta l = l - f(L^0)$, $c_h = -h(L^0)$ and μ is the Lagrangian multiplier (McGlone et al., 2004).

Then the covariance matrix for an unknown 3D edge $\hat{L}$ and the estimated residuals $\hat{v}$ can be obtained

$$C_{\hat{v}\hat{v}} = C_{ll} - AC_{\hat{L}\hat{L}}A^T \tag{4.27}$$

with $$C_{\hat{L}\hat{L}} = M^{-1} - M^{-1}H(H^T M^{-1} H)^{-1} H^T M^{-1} \tag{4.28}$$

and $$M = A^T C_{ll}^{-1} A \tag{4.29}$$

The estimated variance factor $\hat{\sigma}^2$ is given by

$$\hat{\sigma}^2 = \frac{\hat{v}^T C_{ll}^{-1} \hat{v}}{N + H - U} \tag{4.30}$$

Finally, the estimated covariance matrix can be obtained by

$$\hat{C}_{\hat{L}\hat{L}} = \hat{\sigma}^2 C_{\hat{L}\hat{L}} \tag{4.31}$$

The initial value for the 3D edge is the intersection of backwardly projecting a 2D edge from the first and last images. The stopping criterion for iteration is that the changes $\Delta\hat{L}$ to estimation should be less than 1% with respect to their uncertainty or over a maximum iteration value.

If the estimated variance factor $\hat{\sigma}^2$ is larger than a suitable threshold σ_{max}^2, or if the solution does not converge, the estimated 3D edge is rejected.

4.3.2.2 Endpoints Decision

The last part of the algorithm is the computation of the endpoints of the 3D edges. By backwardly projecting rays from the endpoints of one corresponding 2D edge and

taking the intersection with the estimated 3D edge, two intersection points can be calculated. Considering the direction vector of a 3D edge, all the separate intersection points are separated into two groups, as shown in Fig. 4.8. The red circled area shows where the intersection points are. Then a set of endpoint candidates for each 3D edge endpoint can be obtained.

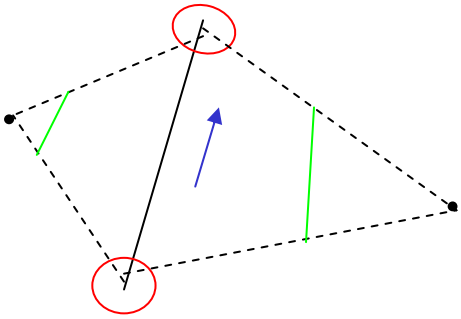


Figure 4.8 Endpoints decision from multiple image edge observations, optical centers (black points), 2D edges (green), 3D edges (black solid line), viewing rays (black dashed lines), direction vector (blue)

The uncertainty value of corrections for each 2D edge is used as a weight for its affection on endpoints of 3D edge. The weight value can be obtained from covariance matrix of estimated residuals. Figure 4.9 shows the edge extraction result on an image of resolution 640×480.

Figure 4.9 Edge extraction result

4.4 Surface patch generation

In this section the method for grouping the extracted 3D points and 3D edges to surface patches is introduced. The main problem in this step is how to recognize feature points and edges that belong to the same surface patch. For example, surface patches 1 and 2 in Fig. 4.10 are in the same infinite plane, but are actually separated. This example shows that coplanarity is a necessary but not sufficient condition for point and edge grouping. Some constraints therefore need to be defined for feature grouping and outline generation in order to find a reasonable surface patch. Only geometric features are

considered during surface patch generation, which means both the grouping and verification are based purely on extracted features.

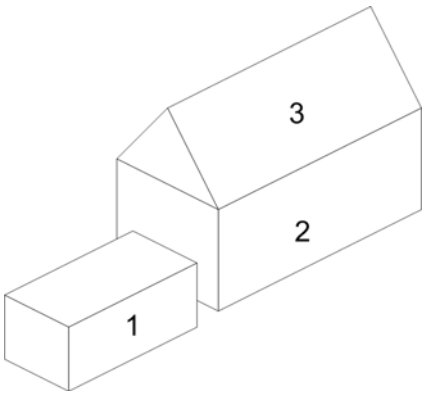


Figure 4.10 Example for different surface patches on the same plane

The method can be divided into five steps: plane hypotheses are formulated from cues based on point-cloud segmentation and on some of the derived 3D edges (section 4.4.1). The hypotheses are then verified and enhanced by incorporating unused 3D edges (4.4.2). Afterwards, plane parameters are obtained through all the edges and points in the plane (4.4.3). After defining the surface patches' outline (4.4.4), the final step is to cull the front face according to their visibility (4.4.5). With the vertical direction and vertical walls, the building's ground plane can be estimated as a by-product of this step. The details are presented in the following section.

4.4.1 Feature grouping

Cues from point cloud segmentation, intersecting edges and parallel edges are considered and applied in sequence. As some points and edges may lie on the boundaries of planes, the proposed grouping method allows for an overlapping clustering result.

4.4.1.1 Point cloud segmentation

Usually the main building facade is the largest plane, with some windows, doors and symbols to let people separate it from other buildings. So it probably contains more salient feature points than other parts of the building. Therefore, point segmentation can help people to recognize the main facades. As shown in Fig. 4.11, two vertical walls can be reconstructed from the result of point cloud segmentation.

The planar surface-growing algorithm by Vosselman et al. (2004) is adopted for point segmentation; it consists of a seed surface detection followed by the actual growing of the seed surface. Since this method is mainly used for laser scanning data, which is much denser than a point cloud extracted from an image sequence without using a dense matching technique, a large surface growing radius is chosen according to the density of points. This leads to some segments that do not exist in the real facade. Only the vertical

point segments are considered as plane hypotheses and wrong hypotheses will be removed in a later step described in section 4.4.2.

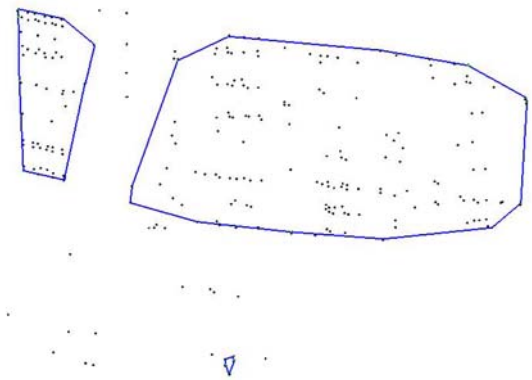


Figure 4.11 Point cloud segmentation result of the data shown in Fig. 4.9, reliable points and point segments with convex outline.

4.4.1.2 Two intersecting edges

If two 3D edges (e_1, e_2) intersect, they must be in the same plane, which may not correspond to the actual building facade. For example, edges 1 and 2 in Fig. 4.12 intersect in object space. Such a plane hypothesis can be avoided by considering the relation between the intersection point and two edges: if the distance between the intersection point and edge is larger than a threshold, the hypothesis is rejected. The distance between the intersection point and finite 3D edge is defined by the relationship between this intersection point and the two endpoints of the edge. If the point is located between two endpoints, the distance value is zero. Otherwise, the distance is the minimal value of the distance between the point and endpoints, which is applicable to edges 1 and 2 in Fig. 4.12. Using the same way of representing knowledge as before, this cue can be written as:

$$\forall e_1, e_2, PlaneFromIntersectingEdges(e_1, e_2) = True \vee False$$

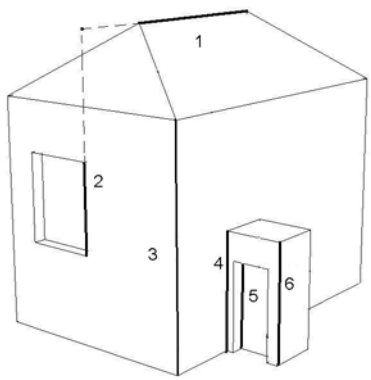


Figure 4.12 A simple building with six extracted edges (1-6)

4.4.1.3 Two parallel edges

There are many parallel or almost parallel edges on the building facade. However, parallel edges that are too close to each other are not reliable for defining a surface patch hypothesis. For example in Fig. 4.12, edges 4 and 5 could result in an unrealistic plane hypothesis, because edge 5 is close to the wall on which edges 4 and 6 are located. A minimal distance between two 3D edges is specified to avoid problems caused by features from detailed parts. The sequence of edge processing in this kind of reasoning should not be arbitrary and additional rules need to be applied. One is that a hypothesis for each edge is first made from the parallel edge with the smallest distance above the minimal threshold. The other is that if either of the two edges was already grouped to a surface patch, other parallel edges belonging to that patch are searched and the two with the smallest distance are chosen to make a hypothesis. For example, edges 3, 4 and 6 are parallel and extracted in that sequence. According to the above rules, edges 3 and 4 form a surface patch before a hypothesis arising from edges 3 and 6. So the plane hypothesis made from the parallel edges 3 and 6 can be avoided. Similar to the cue for the intersecting edges, this can be expressed as:

$$\forall e_1, e_2, PlaneFromParallelEdges(e_1, e_2) = True \vee False$$

4.4.2 Plane verification and enhancement

After defining the plane hypotheses, they are verified and enhanced by 3D edges that were not used for grouping so far. Two parameters must be given first to decide whether an edge belongs to a plane: a threshold ε determining the maximum distance of endpoints from the plane and the maximum angle θ between the edge and the plane. The threshold for the angle is currently a fixed value (10°). Therefore, the relation between a surface patch (s) and an edge (e) is:

$$\forall e, s, EdgeOnPlane(e, s) = True \vee False$$

As the method focuses on main building structures, features within 0.2 m can be ignored and those above 1 m have to be identified, which means two features with a distance between 0.2 m and 1 m can be used to make a plane hypothesis. Therefore, the threshold for parallel edges is 0.2 m. The default value of the maximum distance between intersecting points and edges, as well as for two parallel edges, is 1 m. These maximum distances are also used when checking whether an edge can be added to an edge set (E) during the plane verification:

$$\forall e, E, EdgeWithinMaximumDistance(e, E) = True \vee False$$

Buildings that are more than 1 m apart will therefore not be connected.

The ways to implement cues and to verify plane hypotheses are different according to which kind of cue they are based on. For the first type, i.e. the point cloud based cue, all the points are segmented at one time. A convex hull is calculated for each segment. Then the verification is done by testing whether there are some 3D edges belonging to that plane and located in that region. If there is no edge belonging to it and it is not parallel to the vertical direction, the corresponding plane hypothesis is rejected. Otherwise, a final surface patch is computed from these points and edges.

For the other two edge-based cue types, edges are checked pair-wise. Thus, this step is to grow a small surface, which is generated from two edges, to a bigger one by adding more edges to it. As the method allows an overlapping clustering result, edges near a surface patch boundary can be grouped to more than one surface patch. The implementation of the intersecting edge-based grouping method is given below as a pseudo-code.

Inputs: An array of 3D edges = *edges*

Initialize: $i = 0$, $j = 0$

For i **from** 0 **to** *edges.size* -2

  For j **from** $i+1$ **to** *edges.size* -1

    If $PlaneFromIntersectingEdges(edges[i], edges[j]) == true$ **then**

      Initialize: a new surface patch s, $count = 0$, $subedgeslist = empty$, E is the edge set of edge indexes in subedgelist

      Calculate plane parameters for s

      While $count < edges.size$ and $count \neq i$ and $count \neq j$ **do**

        If $EdgeOnPlane(edges[count], s) == true \wedge EdgeWithinMaximumDistance(edges[count], E) == True$ **then**

          Push $count$ to $subedgelist$

        End if

      End while

      If $subedgelist.size > 0$ **then**

        Push i, j to $subedgelist$

        Reconstruct surface patch outline from edges corresponding to $subedgelist$

        Identify edges near outline and remove their indexes from $subedgelist$

        Remove edges that are corresponded to $subedgelist$ from $edges$

        Push s to surface patch set

      End if

    End if

  End for

End for

4.4.3 Plane parameter estimation

Plane parameters are obtained using all the edges and points in the plane. All possible combinations of two edges are used to define the plane, and remaining points and edges are projected to the plane. The one with the least residual root mean square (RMS) is chosen as the best fit. As some points and edges may lie on the boundary of two planes, the overlapping cluster method results in more surface patches, but it requires an effective outline reconstruction method that can present the patches' shape and correctly judge edges at the boundary.

4.4.4 Outline reconstruction

Through the orientation and position of the surface patch, its semantic information can be concluded and, together with the position and orientation of the edges, a reasonable shape can be derived (defined in section 3.3):

$$\left.\begin{array}{l} Orientation(s_i) \wedge Position(s_i) \Rightarrow Semantic(s_i) \\ Position(E_i) \wedge Orientation(E_i) \end{array}\right\} \Rightarrow Shape(s_i)$$

Using the estimated normal direction, the 3D outline reconstruction problem can be simplified to a 2D problem by rotating the plane into the *XY*-plane. This is possible because surface patches are assumed to be planar. Based on the position and orientation relation between edges and surface, which are decided by corresponding edges and surface coverage in the *XY*-plane in rotated space, the best fitting shape type (defined in section 3.3) is chosen for this surface patch.

4.4.5 Face culling

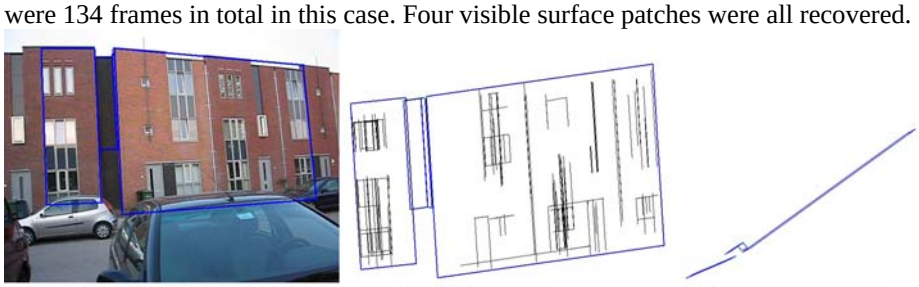
In the above steps, the normal vector of a surface patch can point either to the outside or to the inside of the building. As the further reconstruction method depends on the relation between adjacent surfaces' direction, the normal vector must be homogenously oriented for each surface patch. Since only the surface patches that are visible from the camera positions are generated, the dot product of the plane's normal ($\boldsymbol{N}$) and the viewing direction ($\boldsymbol{C}$) should satisfy the following equation:

$$dot(\boldsymbol{N},\boldsymbol{C}) \geq 0 \tag{4.32}$$

As the sequence of boundary points is consistent with the normal vector in a right-hand coordinate system, the sequence must be modified if the normal vector of a surface patch is changed to the opposite direction.

4.5 Discussion

Figure 4.13 shows the result of the surface patch generation of the building facade shown in Fig. 4.9. The video was captured by a hand-held SONY camera. The images had a resolution of 640×480 pixels and a frame rate of 30 frames per second. There were 134 frames in total in this case. Four visible surface patches were all recovered.



(a) Projection on one frame (b) 3D side view (c) 3D top view

Figure 4.13 Result of surface patch generation of the building facade shown in Fig. 4.9 with the reconstructed surface outlined in blue

Figure 4.14 Feature extraction result, reliable points (green), matched edges (red)

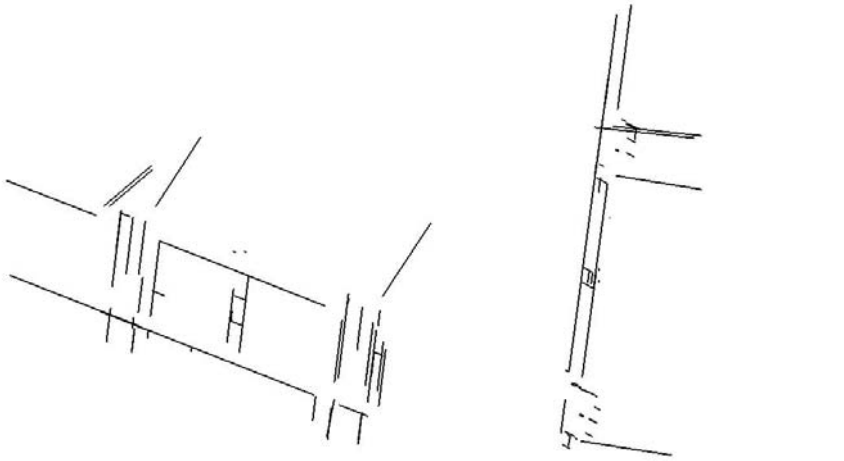


Figure 4.15 3D view of edge extraction result, side view (left), top view (right)

Figure 4.14 shows extracted 3D points and edges projected on one frame of another video image sequence. The video was taken by a camera mounted on a trolley. The camera was oriented sideways and captured the facades of buildings. A 3D view of the edge extraction result is shown in Fig. 4.15. Few edges on the roof were successfully matched, which resulted in only one roof surface patch being generated. The result of the surface patch generation of the building facade is shown in Fig. 4.16. From this, the basic structure for this part of the construction can be recognized. However, outlines of some of the surface patches need to be modified, e.g. there is a long 3D edge on the main wall plane (surface patch 1), which resulted in part of it occluding the other surface patch behind it (surface patch 2). As the edges of the glass wall and door are in the same plane and near to each other, surface patch 2 includes all of them. In addition, one surface patch does not exist in the real world (surface patch 3), because the intersection edge of surface patches 4 and 5 was not observed. Two parallel edges,

which are at the boundary of those two surface patches, form surface patch 3.

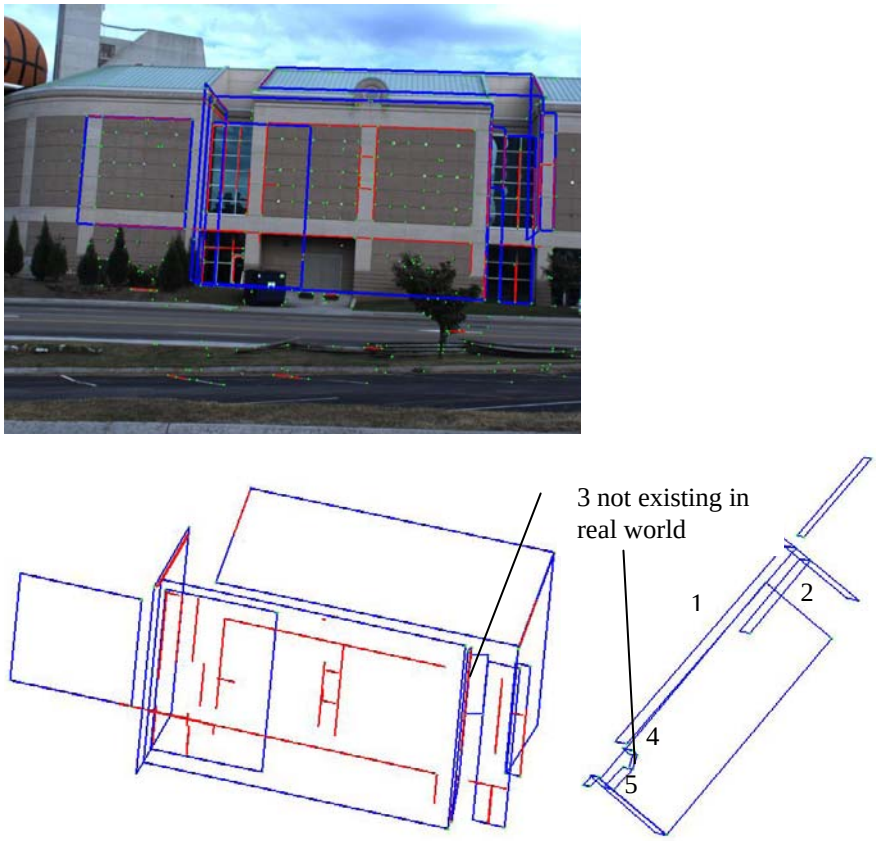


Figure 4.16 Result of surface patch generation of the building facade shown in Figs. 4.14 and 4.15. 2D view (top), 3D side view (bottom left), 3D top view (bottom right), reliable points (green), extracted edges (red), surface patches (blue)

In this chapter, feature extraction and grouping methods are described. Due to the limitations of image quality and imperfect feature extraction results, some edges cannot be extracted from images or easily broken into several segments. Also, due to the disadvantage of a short baseline, 3D edges that have high geometrical accuracy are considered as successfully matched edges. As only sparse points and edges are obtained and some semantic edges are missing or not complete, extracted surface patches are located in a correct, infinite plane. However, most of the time, they do need to be extended or broken into several small parts.

5 Polyhedral model reconstruction

An image-based reconstruction method for modeling buildings should be able to determine the model representation for buildings efficiently from image interpretation. The first problem that needs to be considered is how to define an appropriate model for the buildings. Large variations in the geometric and functional descriptions of the buildings make the model definition a difficult problem. Another problem is that buildings may be partly occluded by other objects or by their own design. In addition, the quality of available images may be very low and the imaging process may introduce many different kinds of noise.

This chapter describes a new method for reconstructing polyhedral objects, which this thesis uses as a generic building model. Building facades are recovered based on features that have been extracted and grouped from terrestrial video image sequences. A hybrid model- and data-based method is used to reconstruct a building model from both extracted surface patches and hypothesized parts. Using knowledge on building structure leads to a simple and fast reconstruction method, and also enables us to obtain the main structures of buildings.

Chapter 4 introduced the main aspects of feature extraction and grouping tasks. The purpose and strategy used in low level processes to detect and extract the surface patches which have a meaningful correspondence with the building facades were discussed. Consequently, it is time to move on to the model presentation of buildings, which was carried out in the reconstruction part of this study. In the proposed approach, the building model reconstruction can be seen as a process of connecting generated surface patches. The perfect situation is achieved when all the surface patches of the building are correctly recovered. However, there are always some parts that are missing or wrong due to occlusions, or a failed or wrong extraction. Meanwhile, some building edges are not represented by salient image edges. As the surface patch generation method is based on extracted point and edge features, the pure geometric reconstruction fails to determine some surface patches and the outlines of other patches may need modifying. Topological constraints help us to make hypotheses on occluded areas and to define relations between different surface patches. The reconstruction procedure therefore consists of different intermediate and inter-related processes, aimed at forming a framework in such a way that every process provides more abstract information and more building structure-related information for the building reconstruction. This chapter is split into three parts: the first part describes some related aspects for the reconstruction method, which include the relations between extracted surface patches and realistic building facades (section 5.1.1), the surface patch class that is defined for implementation (section 5.1.2) and the intensity-based surface patch verification method (section 5.2). These are followed by part two, the building model reconstruction method. Surface patch neighborhood relationships are set up first (section 5.3). The local model hypotheses are then drawn up based on adjacent surface patches (section 5.4). Afterwards, all the local models are connected to form a complete building model (section 5.5). In section 5.6, the model refinement is described to improve the

reconstruction result. Finally, in part three, texture mapping, which aims to make the reconstructed model more realistic, is described (section 5.7).

5.1 Primary facade elements

The objects described in this thesis are common, man-made buildings with planar structures, which are different from specific architectures. Building structure knowledge therefore is able to be selected for the reconstruction process. The initial knowledge about the semantic information and outline shape has already been entered into the surface patch generation step (see section 4.4). In fact, this knowledge reduces the complexity of making surface patch hypotheses and provides reliable guidance in recovering the shape of surface patches. The perfect situation is achieved when all the surface patches of the building are correctly recovered. However, due to the reasons discussed in section 4.5, some extracted surface patches do not completely fit the building facades or roofs. The relation between extracted surface patches and the actual building structure elements therefore needs to be discussed. This will be described in section 5.1.1.

There is another issue that needs to be discussed before describing the modeling steps. As mentioned above, a hybrid model- and data-based method is used in this thesis and the topological relations determined between the surface patches are used as important information during the reconstruction steps. Thus, the surface patch should include some parameters to represent these topological relations and how to use them efficiently during the reconstruction steps. More details can be found in section 5.1.2.

5.1.1 Relation between surface patches and building structure elements

It was described in chapter 4 that how surface patches can be generated based on grouping feature points and edges that have been extracted from image sequences. The surface patches are generated based on geometric grouping and hypothesis, so their general plane parameters are correct. However, the building facades and roofs are regions with boundaries. On the one hand, these boundaries may be irregular, due to the design, intrusions or extrusions, while on the other hand, sparse points and edges cannot guarantee that the boundary features can be extracted. More reasons why there may be some problems in the results from surface patch generation were discussed in section 4.5. Since the further model reconstruction is based on the surface patches, the relation between the surface patches and the building structure elements should be discussed in more detail.

Some common relations between generated surface patches and realistic building structure elements are shown in Figs. 5.1 and 5.2. They can be divided into two types: the first type is a surface patch that covers more area than in reality (over-covered). As shown in Fig. 5.1, it is possible that two real building elements are located in the same infinite plane, or that one edge of an element is located in the same infinite plane as another element. Therefore, the model method should be able to separate an initial surface patch and make a hypothesis about the location of new surface patches if there

is evidence they may exist. The second type is a surface patch that covers less area than in reality (under-covered) (Fig. 5.2). It seems easy to deal with this by using intersecting adjacent surface patches, but then the problem becomes how to identify the adjacent surface patches correctly even when they are far away from each other.

Figure 5.1 Example of a surface patch that covers more area than in reality

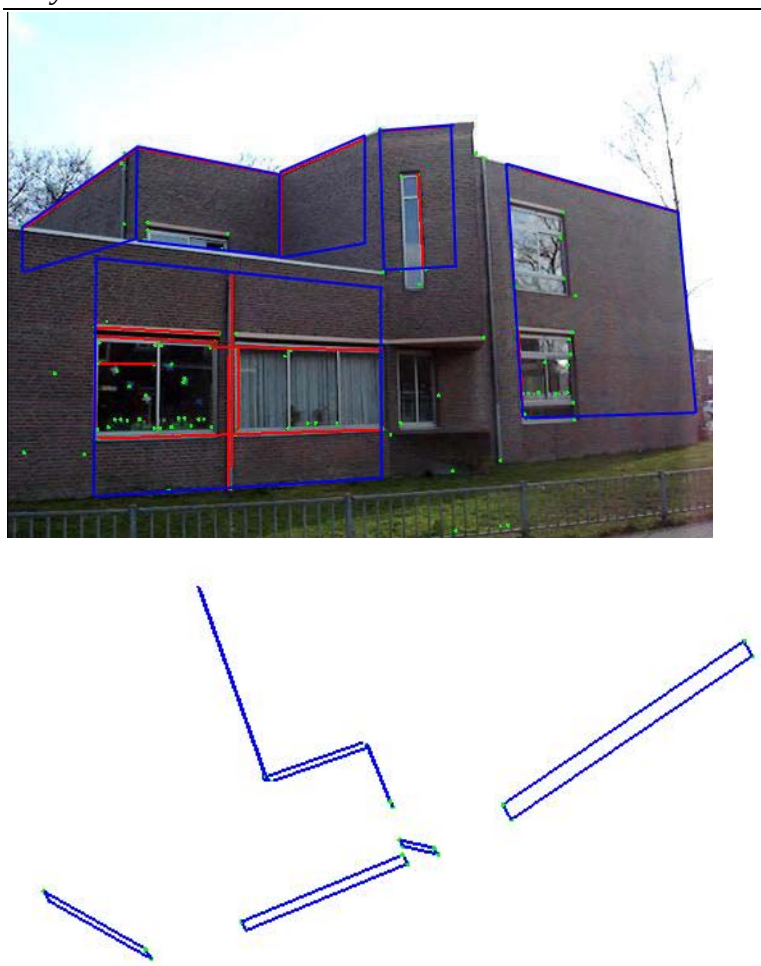


Figure 5.2 Example of surface patches that cover less area than in reality

5.1.2 Surface patch presentation

A surface patch is primarily a plane, which is represented by the normal vector and the distance of the plane to origin. However, a surface patch is also a limited region. The vertex can be used to record the boundary for polygon shapes. As mentioned in section 4.4.5, the surface patches in this thesis must point to the outside of the building. Therefore, the boundary line topology should be in a certain sequence. Here, the sequence is consistent with the normal vector in a right-hand coordinate system. There are three surface patches in Fig. 5.3, the point coordinates and line topologies can be recorded separately, in order to easily change the sequence of boundary points. The boundaries of these surface patches can be presented as shown in Table 5.1. The starting point of a line topology can be any point, but it must be the same as the last point to form a closed polygon. So, two points with contiguous numbers in line topology represent a boundary edge.

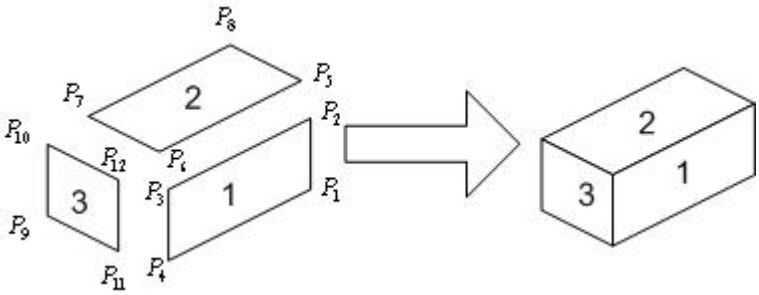


Figure 5.3 Surface patches (left) and the local model based on them (right)

Table 5.1 Surface patch boundary presentation

Surface number	Boundary points	Line topology
1	P_1, P_2, P_3, P_4	1,2,3,4,1
2	P_5, P_6, P_7, P_8	8,7,6,5,8
3	$P_9, P_{10}, P_{11}, P_{12}$	9,11,12,10,9

If the three surface patches in Fig. 5.3 can form a local model, some parameters are needed to record the relation between adjacent surface patches. In order to easily describe the topological relation and deal with further verification steps, the corresponding edges of two adjacent surface patches should refer to each other by the surface and point number. Since each edge has a fixed direction, its first point number is used in this thesis as the label for the whole edge. Based on each boundary edge, the adjacency relationships of surface patches can be shown as:

$$SurfaceID \times 1000 + FirstPoint\ Number$$

Topological relations follow the same sequence as line topology and the default value for surface ID and first point number is zero. Some other information of a surface patch, such as its semantic meaning, extracted points and edges on it and visible frame range for it, are also important characteristics in the Surface class below.

```
Class Surface
{
    int id;                ///id of the surface patch
    int label;             ///semantic type of the surface patch
    int firstframe, lastframe;            ///visibility in frame range
    PointNumberList  pointnumbers;   ///3D points on the surface
    LineNumberList  linenumbers;      ///3D edges on the surface

    Vector3D normal;       ///Normal vector
    Double distance;       ///Distance of plane to origin
    ObjectPoints  surfacepoints;        ///surface boundary points
    LineTopology  surfacetop;           ///surface boundary topology
    PointNumberList   connectpoints;  ///the first point number of each connected edge
from adjacent surface patch
}
```

5.2 Surface patch verification

If a surface patch is visible in the images, a high cross-correlation result can be calculated for the particular areas that are visible in any two images. So the similarity over the image sequence is computed to evaluate the surface patch, which is similar to the approach by (Baillard and Zisserman, 1999). Instead of computing the correlation with respect to the whole surface, only the endpoints from edges are considered and computed for the respective similarity within a 5x5 window. This processing is motivated by the observation that whole surfaces may be partly invisible in one frame, e.g. because of occlusion, thereby biasing the correlation and making an overall verification difficult.

In more detail, given the plane π, there is a homography represented by a 3×3 matrix H_i between the first and i th frame, so that corresponding points are mapped as

$$x_i = H_i x_0 \tag{5.1}$$

where x_0 and x_i are image points corresponding to the same object point X and represented by homogeneous 3-vectors.

The homography matrix is obtained from 3×4 camera projection matrices for each frame. For example, if the projection matrices for the first and i th frames are $P_0 = [\, I \,|\, 0 \,]$ and $P_i = [\, A \,|\, a \,]$, and a plane is defined by $\pi^T X = 0$ with $\pi = (v^T, 1)^T$, then the homography induced by the plane is:

$$H_i = A - a v^T \tag{5.2}$$

In this thesis, the first frame in which the surface patch is visible is chosen as the reference frame and the cross-correlation is computed between the reference frame and other frames within the frame visibility range. The homography for $x_i = H_{ri} x_r$ is:

$$H_{ri} = H_i H_r^T \tag{5.3}$$

where i represents the i th frame and r the reference frame.

The points of interest are the endpoints of the 2D edge extraction result within the projected surface area in the reference frame. Points that are regularly distributed over that image area are also chosen. The similarity score for the average, normalized cross-correlation value (Eq. 2.1) for points ($x_{rj}, j = 0, \cdots, m$) in the valid images (e.g. n+1 frames in total) is:

$$sim = \sum_{j=0}^{m} \left(\sum_{i=1}^{n} NCC^2 (x_{rj}, H_{ri} x_{rj}) / n \right) / (m+1) \tag{5.4}$$

The normalized cross-correlation is squared to keep its range in [0,1] and to give more weight to high scores.

Over-covered surface patches contain areas that do not belong to the surface patches. Therefore, the similarity score as an average value is affected. Even under-covered surface patches may not gain a high similarity score, because there are intrusions or extrusions on the main surface patches, or they may be occluded by other parts of the building. To test the similarity scores of extracted surface patches, this verification method has been applied to over 100 surface patches that were generated from the previous section. Almost all the surface patches can gain a score higher than 0.6. In order to guarantee the accuracy of surface hypotheses made during local model reconstruction and visible in the image sequences, this verification method is adopted and the hypothesis is accepted only if the similarity score is higher than the threshold (set at 0.8 by default).

5.3 Searching the neighboring surface patches

From analyzing results of the relation between surface patches and realistic building structure elements in section 5.1.1, two neighboring surface patches may not have boundaries that are near to each other. Some rules therefore have to be chosen for selecting the correct surface patches for computer implementation. After intersecting neighboring surface patches, some patches can be separated into several parts or extended. Their boundaries can be changed within reasonable limits. Finally, the searching sequence is described in the following sections.

5.3.1 Rules for selecting neighboring surface patches

As introduced in the related concepts section (section 3.2), there are two kinds of neighboring relations:

$Relation(s_1, s_2) = Meet \vee Attach$

According to the definition, the common characters for neighboring surface patches are that they are adjacent and intersecting, which means the distance between them is zero and the intersecting line is at least along one surface patch boundary. When intersecting two surface patches, a finite intersecting line l and four intersecting points p_1', p_2', p_3' and p_4' can be obtained as shown in Fig. 5.4.

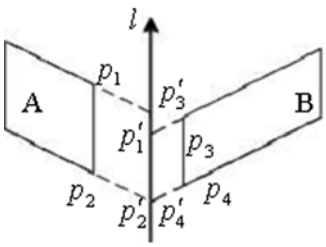


Figure 5.4 Two intersecting surface patches

Points p_1 and p_2 are nearest the boundary points of p_1' and p_2' in the left surface patch. Similarly, points p_3 and p_4 are nearest the boundary points of p_3' and p_4' in the right surface patch. If the distance d between p_i and p_i' is large, a surface extension may not be reliable; by default the extension should not be larger than the extracted surface patch. So this distance d is one parameter used to evaluate the surface patch found.

$$d = (\sum_{i=1}^{4} | p_i - p_i' |) / 4 \tag{5.5}$$

From Fig. 5.4, it is obvious that edge $l_{(p_1',p_2')}$ and edge $l_{(p_3',p_4')}$ must have some overlap. The percentage of overlap on the intersecting edge can be used as a parameter. Since the intersecting line (l) can be defined by a foot point (p_f) and a direction vector (n), each point can be presented by the corresponding scalar (t).

$$l = p_f + n * t \tag{5.6}$$

$t_1, \cdots, t_4$ are defined as scalars associated to points $p_1', \cdots, p_4'$ and let t_1 be larger than t_2 and t_3 be larger than t_4 .Then points corresponding to $min(t_2, t_4)$ and $max(t_1, t_3)$ define the intersecting edge. Therefore,

$$\rho = \begin{cases} 0, & if \quad t_1 \le t_4 \quad or \quad t_2 \ge t_3 \\ (min(t_1,t_3) - max(t_2,t_4))/(max(t_1,t_3) - min(t_2,t_4)), & else \end{cases} \tag{5.7}$$

where ρ is the overlapping ratio and $\rho \in [0,1]$.

According to preferential knowledge (as defined in section 2.2), a building ground plane is mostly composed of rectangles. Thus, the angle $\theta \in [0^o, 180^o]$ between two surface patches is also considered. Obviously, if two surface patches are parallel or have no overlap, they cannot be adjacent. For other cases, if the value of d (Eq. (5.5)) is very small, e.g. smaller than the standard deviation of distance for extracted features to the plane they are grouping to, the adjacency can be ensured. So a small distance may imply a higher probability. The three similarity measures are combined as:

$$\varepsilon = f(d, \rho, \theta) = \rho * f(\theta) * f(d) \tag{5.8}$$

Different weights can be assigned to these three parameters, and simple ways are chosen in this thesis: $f(\theta) = sin(\theta) \in [0,1]$ and $f(d) = d^{-1} \in (0, +\infty)$. Therefore, $\varepsilon \in [0, +\infty)$ and the valid neighboring surface patches must have a value larger than zero. The surface patch with the largest ε is considered to be adjacent to the current surface patch.

5.3.2 Surface patch modification

By searching along surface patches based on the above rules, two adjacent surface patches A and B can be found. So,

$Adjacent(s_A, s_B) = True$

To be sure they are valid neighboring surface patches, the intersection type has to be checked by the location of intersecting lines. Meanwhile, as mentioned before, some surface patches may be over- or under-covered, and surface patch modification should be done when it is valid. Some examples for two intersecting surface patches are shown in Fig. 5.5.

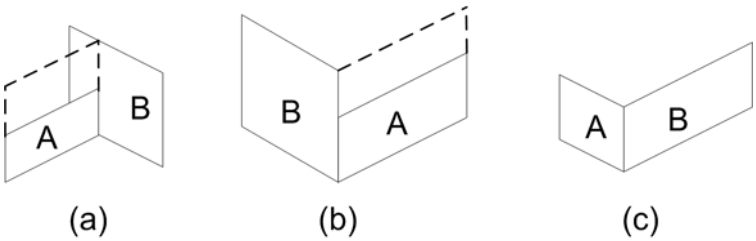


Figure 5.5 Example of two intersecting surface patches. (a) The intersecting line is at the boundary of one surface patch and contained within another one, (b) the intersecting line is at the boundary of both surface patches but the intersecting points are not consistent, (c) the intersecting line is at the boundary of both surface patches and the intersecting points are consistent

If surface patches belong to a local model, intersecting points must be consistent. Since the generation result may be imperfect, a small threshold (0.2 m by default) is allowed. After modifying the intersecting points to make them the same as the endpoints of the defined intersecting edge, surface patches in Fig. 5.5c can be presented as:

$Intersection(s_A, s_B) = ShareBoundaryEdge$.

In Fig. 5.5a, the intersecting line is contained in surface patch B and the intersecting points are not consistent. The verification method is used to check whether an extension part (shown as a dashed boundary) of surface patch A is valid or not. If such an extension is accepted, surface patch B will be divided into two parts B1 and B2. They have the same infinite plane parameters, but different boundary points and line topologies. The intersection relation between B1 or B2 and A is the same. Clearly, only one of them is coherent with surface patch A to form a local model. The coherence will be checked later.

$Intersection(s_A, s_{B1}) = ShareBoundaryEdge$

$Intersection(s_A, s_{B2}) = ShareBoundaryEdge$

If the extension is invalid, the intersection relation between surface patches A and B is $Intersection(s_A, s_B) = NotShareBoundaryEdge$. Therefore, their neighborhood relation is surface patch A is attached to surface patch B: $Relation(s_A, s_B) = Attach$

In Fig. 5.5b, the intersecting line is at the boundary of both surface patches but the intersecting points are not consistent. Similar to the above case, an extension needs to be verified. If the extension is valid, the situation is the same as that in Fig. 5.5c. Otherwise, the situation is more complex and has more than one possibility. By default, surface patch A is attached to surface patch B. However, their semantic meaning and their topological relations with other surface patches could make it easier to decide on the relation between A and B. The sequence of searching for neighboring surface patches is discussed in the next section.

5.3.3 Processing sequence

The basic idea in geometric modeling is to combine simple shapes to construct complete models. In 3D building reconstruction, as studied in this thesis, the adjacent 3D planar polygons are combined to construct the facade structure with part-roof structures. Topological properties are not metrical, but concern such things as connectivity and spatial continuity. The concept of spatial adjacency, which is normally defined based on a point-cloud data set, is extended here to sparse polygons. Rules for searching for neighboring surface patches are defined above, for when they are not directly connected with each other.

Clearly, if two surface patches contain the same edge, they must be adjacent. Based on the overlapping clustering method, such a situation can exist and those surface patches are dealt with first. After that other neighboring surface patches are searched for along boundary edges based on the above method. The characteristics of input data are also considered in the searching sequence.

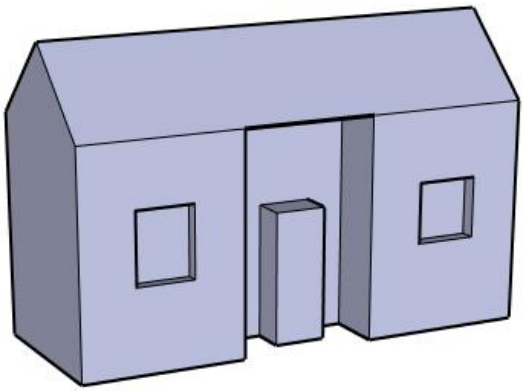


Figure 5.6 Example of two walls connected by a roof

According to the semantic classifications in this thesis, surface patches are divided into wall, roof, ground plane and extrusion. These, except for ground plane, are considered in this step. Walls and extrusions are difficult to separate according to only the surface patch generation result. Roofs are not well observed by a terrestrial video image sequence, which leads to fewer features being extracted from roofs than from walls and extrusions. Therefore, when two roofs locate in the same infinite plane, the chances they are over-covered are high. However, observed from the ground, the relation between the

roof and wall is simpler than that between walls, and some walls may intersect with the same roof (Fig. 5.6). Roof surface patches are therefore dealt with before other surface patches.

5.4 Constructing a local model

Local models contain three surface patches and they are usually constructed from two surface patches with the first kind of neighboring relation, $\forall s_1, s_2, Relation(s_1, s_2) = Meet$ (defined in section 3.2). The third patch is searched for during reconstruction, and the preferential knowledge provides essential guidance when there are not enough features extracted to make a decision.

From the above steps, two surface patches with consistent intersecting points can be found. Another requirement for valid neighboring surface patches, the coherence, still needs to be checked.

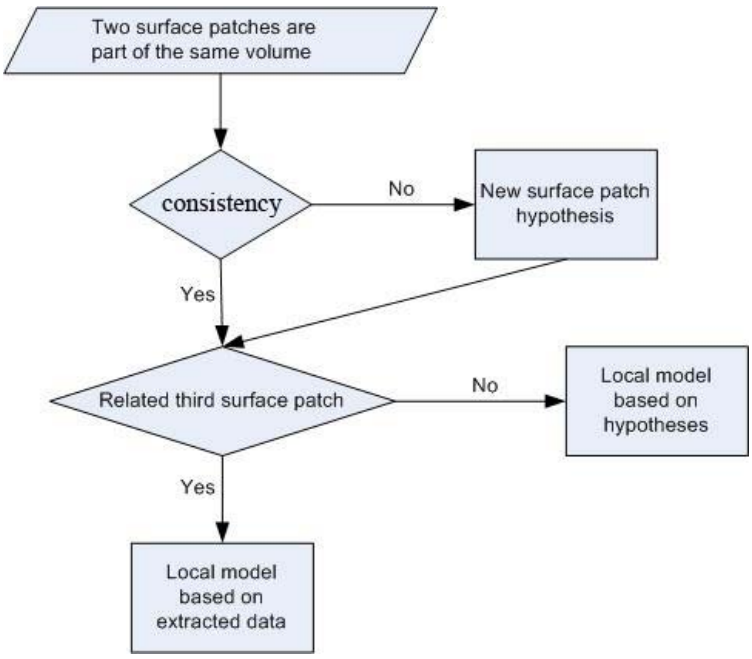


Figure 5.7 Workflow for constructing a local model

Each local model is formed by three adjacent surface patches. If they are all extracted, a local model can be formed by intersecting them. However, mostly, only two of them can be observed, which leads to the constructing method starting from two adjacent surface patches. Since generated surface patches may be modified during a local model reconstruction and they can be used to reconstruct other local models, two rules are defined:

1) A local model has to be reconstructed from at least two generated surface patches.

2) If a surface patch has been extended by more than 50% coverage during modification, no further hypothesis can be made based on it.

As can be seen in the workflow shown in Fig. 5.7, new surface patches are made and verified if neighboring surface patches are not coherent. Such a non-coherent situation is usually caused by over-grouping features from surface patches with similar plane parameters. So, if two nearby surface patches are not coherent, the boundary of one surface patch needs to be modified and a new surface patch hypothesis can be made, based on these two surface patches, to form a suitable local model.

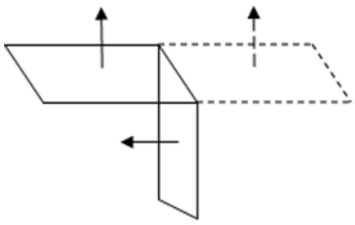


Figure 5.8 Example of new surface patch hypothesis

Figure 5.8 shows one example of how to make a new surface patch hypothesis. A surface patch is extended across a common edge to make a new hypothesis. Therefore, the new surface patch and the unmodified surface patch must be coherent. The verification method given in section 5.2 is then applied to this new surface patch. If the surface hypothesis is accepted, the local model hypothesis can be made.

Figure 5.9 Example of local model hypotheses

After two coherent, adjacent, surface patches are found, both their topological relation and a local model can be built. During this step, a surface patch to support the hypothesis is searched for first. The searching step is similar to the earlier step for searching for adjacent surface patches, and here two boundary edges should be considered. If three surface patches that form a local model are found, the local model is defined by them. Otherwise, simple block types based on building structures as defined before are chosen to fit them, e.g. as shown in Fig. 5.9. Hypotheses are not applied to surface patches that are not along the streets. As building ground mostly has rectangular angles, a vertical wall is assumed to be perpendicular to its neighboring wall when it is self-occluding. The above two sentences can be presented as an example below:

$$\exists s_1, Semantic(s_1) = Roof, \quad \exists s_2, Semantic(s_2) = Wall \quad \Rightarrow$$
$$if \quad Relation(s_1, s_2) = Meet \quad \wedge \quad \neg(\exists s, Relation(s, s_1) = Meet \wedge Relation(s, s_2) = Meet), \quad then$$
$$\{s\} = \{s\} + s_n \wedge Angle(s_n, s_2) = RightAngle \wedge Relation(s_n, s_1) = Meet \wedge Relation(s_n, s_2) = Meet$$

If the third plane is visible in the image sequence, the verification test is used. For example, some roofs may be visible from the ground; the possibility of an oblique roof is tested for. Otherwise, the third surface patch is estimated based on common building structures. It will be reasonable, but could be inaccurate.

Another general hypothesis is that a roof is horizontal if there is no extracted feature on the roof. When two coherent, vertical, surface patches and some other features above them are found, the plane sweeping method is used to find a more reliable roof plane. From the node that will have degree three after local model generation, the two non-vertical edges can form the initial roof plane hypothesis. If the initial roof hypothesis is rejected, a new roof hypothesis is made by rotating the original polygon around its horizontal edge. The optimal angle is computed by searching for the maximum of function sim (cf. Eq. 5.4) over a range $[-\frac{\pi}{6}, \frac{\pi}{6}]$ with 1° each time. The roof position is indicated by the hypothesis with the maximum similarity score.

5.5 Connecting separate local models

During local model construction, the topological relationship between surface patches is established as well. However, a surface patch is only related to two surface patches at one time during the local model construction. Relations between neighboring surface patches are recorded by each edge's connecting parameter and the whole building model is reconstructed by connecting local models one by one. If the building belongs to simple block types, the building model has already been recovered during the local model construction step. Figure 5.10 shows an example in which the arrow shows the normal vector of surface patches 1, 2 and 3. According to our sequence, surface patch 1's neighboring surface patches are searched for first. There are two possible middle cases, but the final results are the same. In the top row, surface patch 2 is found before surface patch 3. Therefore, a local model with corner point P_1 is recovered together with the fact that surface patches 1 and 3 are valid neighboring surface patches. Then surface patch 4 is estimated by hypothesis to construct the local model with a corner point P_2. In the bottom row, surface patch 3 is found before surface patch 2. The local model with corner point P_2 is recovered before the one with corner point P_1.

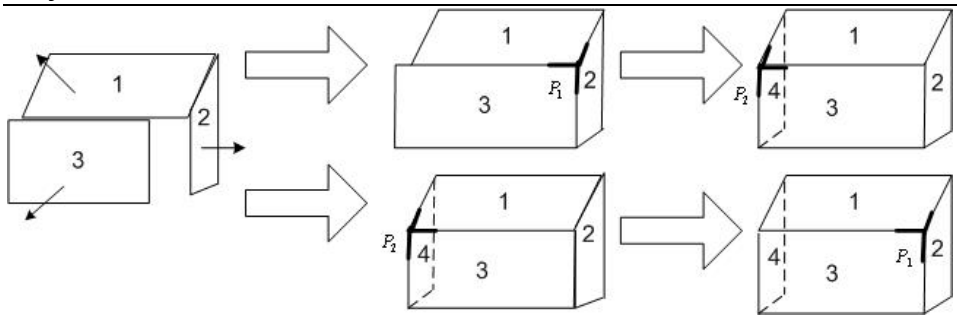


Figure 5.10 Example of connecting simple local models

For complex buildings, there are always some surface patches that are attached to surface patches that are part of a different local model. They usually belong to extrusions and intrusions of the buildings. Figure 5.11 shows two examples. Two edges of surface patch 1 in (a) are contained in different surface patches (2 and 3). In (b) surface patch 4 belongs to the same local model as surface patch 5, and one of its edges is contained in surface patch 6. These surface patches connect separate local models and are generalized by simple block types based on the assumption that all connected surface patches belong to one building. For example, surface patch 1 attaches to surface patches 2 and 3, which should be connected in one building. To set up their relations, the extension of surface patch 1 has already been checked and rejected. Therefore, as shown in (c), surface patch 1 is assumed to be part of a block together with surface patch 8. Then surface patch 7 is estimated to connect surface patches 2 and 3, which needs to be checked by the surface patch verification method. Surface patch 9 in (d) belongs to the same local model as surface patches 4 and 5. As surface patch 4 is attached to surface patch 6, surface patch 9 should also be attached to surface patch 6 and points P_0 and P_1 must locate in surface patch 6. Thus, the relations between four surface patches in this building can be built up.

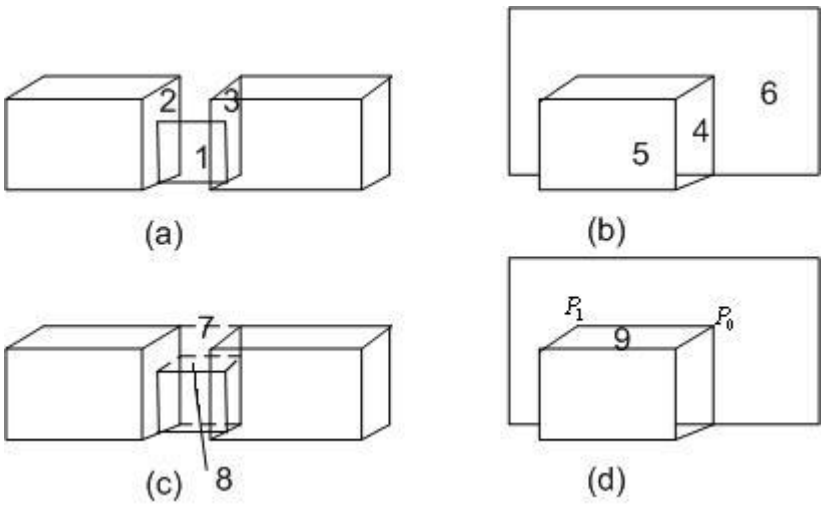


Figure 5.11 Example of connecting separate local models

5.6 Model Refinement

Building model refinement can improve a coarse building model or add more detailed models to the original building model. To improve the quality of coarse building models involves designing a way to determine a reliable and accurate geometric description of the 3D structural elements of a reconstructed coarse model. Most of the time, it is a problem about how to find the correct boundary for each plane of the coarse model. This can be tackled in an interactive way, using redundant images or fusing different kinds of data. For adding more detail structures to a generalized coarse model, the main problem is how to locate the detailed structures and recognize their structure. Details can be found and estimated from features that do not correspond to the coarse model. Besides geometric features, the texture can also be used to identify the parameterized components, such as windows and doors.

Due to the quality limitation of input video image sequences, details are difficult to be correctly identified from image based on geometric features. Meanwhile, the visibility of building facades from ground limits the possibility to reconstruct the complete building models. Only the following two aspects are considered to refine coarse building models based on input data.

5.6.1 Model constraints

According to the definition of a local model, it is constructed by three adjacent surface patches. However, sometimes a local model is formed by four surface patches, such as the example in Fig. 5.12. The difference between this case and normal local models is that these four surface patches intersect at one corner point P and this case has to be confirmed by surface extraction results. If two walls are extracted, the position of P is decided by them and roofs are adjusted. Otherwise the position of P is decided by the first two surface patches used to make this local model hypothesis, and other surface patches are adjusted. If only surface patches 1 and 2, or 3 and 4, are visible in the image sequence, it is difficult to make a local model hypothesis based on them which does not satisfy the basic block types. Since walls should be vertical for the buildings described in this thesis, surface patches 2 and 4 are enough to recover the local model when they are roofs. Otherwise, this kind of local model can also be recovered when more than two adjacent surface patches are found.

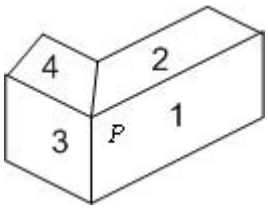


Figure 5.12 A local model with four adjacent surface patches

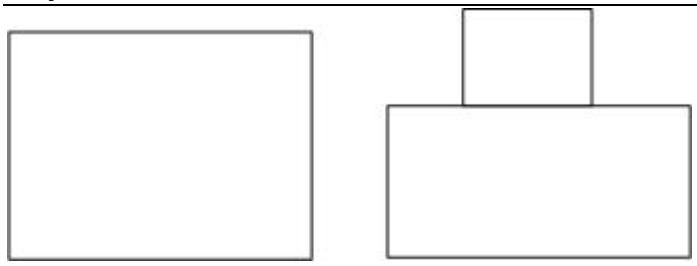


Figure 5.13 Buildings with rectangular ground plane

Figure 5.14 Orthogonal image with building map outlines (images: © Blom)

Occlusion is the problem that cannot be avoid with terrestrial image sequences. Objects in front of buildings, such as people or trees, may lead to only part of the building facades being visible in the image sequence and this may lead to surface extraction failure and unreliable verification results. However, buildings with a rectangular ground plane or a combination of rectangles, as in Fig. 5.13, are common in the real world (Fig. 5.14). When four adjacent walls can form a box-type building, they are connected based on such model constraints as given below:

$$\exists s_1, s_2, s_3, s_4, Semantic(s_1) = Semantic(s_2) = Semantic(s_3) = Semantic(s_4) = Wall \wedge Relation(s_1, s_2)$$
$$= Relation(s_2, s_3) = Relation(s_3, s_4) = Meet \wedge Angle(s_1, s_2) = Angle(s_2, s_3) = Angle(s_3, s_4) = RightAngle$$
$$\Rightarrow if \quad Normal(s_2) \cdot Normal(s_4) < 0 \quad then \quad Relation(s_1, s_4) = Meet$$

As an extension of the preferential knowledge that building ground planes mostly have rectangular angles, the orthogonal constraints enable the method to recover some simple building structures when they are partly occluded.

5.6.2 Roof improvement

As buildings in this thesis are restricted to some standard types, one specification is that roofs intersect with walls. However, building wall outlines are usually different from roof outlines because of overhangs, etc. Figure 5.15 shows three common types of building roofs with overhang. A flat roof cannot be observed from ground-level, so only oblique roofs are checked.

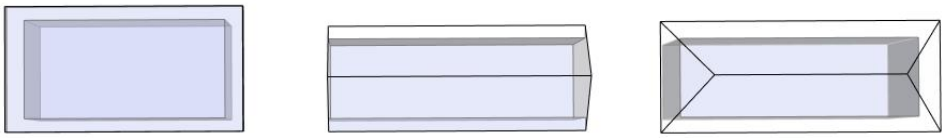


Figure 5.15 Top view of building roofs with overhang (flat, gable and hip roof)

When roofs and walls are well estimated, the intersecting line or original roof boundary edge is on the roof. Roofs can be improved by searching for extracted edges in the roof extension area (Fig. 5.16). Therefore, the lower boundary edge of a roof is projected into one image and searching can extract 2D edges in the roof extension area. If there are edges parallel to the original projection edge, new edges are calculated by the intersecting plane formed by the camera position and the 2D edge with the roof plane. Then the new edge is projected to another two images for verification. Figure 5.17 shows an improved roof result.

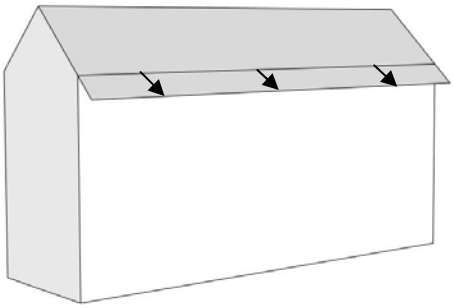


Figure 5.16 The extension direction for roofs

Figure 5.17 The result of roof improvement

5.7 Texture mapping

When the quality of natural images cannot satisfy the requirement for texture mapping, building models can be textured with predefined images, such as artificial images. For example, walls can be textured with a repeated brick image and roofs can be textured with an image with a predefined color. However, this kind of images cannot represent the real building facades and reconstructed models are always restricted to a limited level of detail. Using the real images as textures can make building models more realistic and virtually compensate for the lack of detail in some models.

In the previous sections, the corresponding projection matrices were recovered and building models were reconstructed from image sequences. The appropriate polygonal image regions that correspond to building model surfaces can be found by projecting the building facades to images. Texturing the building models is a reverse process: the image regions are warped to remove projective distortion and then color values within the polygon are assigned for each pixel in the texture. It seems a straightforward step, but it still includes texture coordinate calculations and texture generation problems. Corresponding texture coordinates have to be calculated for each vertex of a polygon on the 3D surface. The transformations between texture and image sequences therefore have to be known. Meanwhile, textures can be generated from image sequences in different ways. The following two sections will describe these. However, detecting and mending highly occluded areas for texture mapping is beyond the scope of this thesis.

5.7.1 Transform from image to texture

Texture is a special kind of image, which aims to be glued on to a surface. It should be the frontal orthogonal view of the surface, as ‘image2’ in Fig. 5.18. However, most of the time, images cannot be satisfied with such a requirement. Under a central projection, shapes become distorted. Projective transformation has to be applied to images, such as ‘image1’, to allow them to be used as textures.

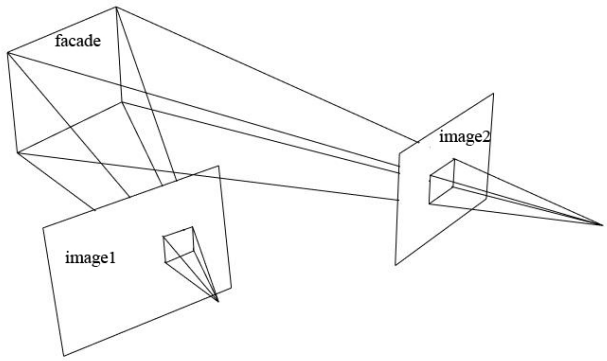


Figure 5.18 Projective transformation

Computation of a projective transformation from point-to-point correspondences has already been introduced in the surface patch verification step (section 5.2). The homography matrix H for equation 5.1 was obtained from projection matrices for each frame. The information of the projection matrix for texture can be replaced by the geometry of planar facades. In the outline reconstruction step, the 3D problem was simplified to a 2D problem by rotating the plane into the *XY*-plane. The corresponding points in the rotated *XY*-plane and texture only have a linear relation. Therefore, each pixel in the texture can easily be set to correspond to a 3D point on the surface patch. Polygon vertices are recorded for the further texture mapping process, which is based on OpenGL. The steps are given below:

1. Rotate each boundary point $\boldsymbol{X}_{boundary}(X,Y,Z)$ of the surface patch s into the reference plane $\boldsymbol{X}_{refernce}(x_{reference}, y_{reference}, z)$, of which the Z axis is the same as the normal vector of the surface patch. After the rotation, all the points have the same z value along the Z axis,

$$\boldsymbol{X}_{reference} = \boldsymbol{R}\boldsymbol{X}_{boundary} \tag{5.9}$$

2. Calculate XY ranges for the surface patch from all the boundary points on the reference plane. The texture size ($W \times H$) can be decided based on four values ($x_{min}, x_{max}, y_{min}, y_{max}$) as below:

$$W = (x_{max} - x_{min}) \times scale \tag{5.10}$$

$$H = (y_{max} - y_{min}) \times scale \tag{5.11}$$

3. In the OpenGL, a texture is defined in its own u, v coordinate system. And the range for u, v is [0,1]. So the vertex coordinates for surface patch polygon are:

$$u = (x_{boundary} - x_{min}) / (x_{max} - x_{min}) \tag{5.12}$$

$$v = (y_{boundary} - y_{min}) / (y_{max} - y_{min}) \tag{5.13}$$

4. If the color value of texture pixel $\boldsymbol{x}_{texture}(x_{texture}, y_{texture})$ should be obtained from image I, there are two steps. The first one is to find the corresponding 3D point $\boldsymbol{X}(X, Y, Z)$ on the surface patch.

$$x = x_{texture} / scale + x_{min} \tag{5.14}$$

$$y = y_{texture} / scale + y_{min} \tag{5.15}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \boldsymbol{R}^T \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{5.16}$$

Then, projecting the 3D point to image I, the point $\boldsymbol{x}_{image}(x_{image}, y_{image})$ can be decided by combing Eq. 4.8 with Eq. 5.16.

$$\begin{pmatrix} x_{image} \\ y_{image} \\ 1 \end{pmatrix} = \boldsymbol{P}\boldsymbol{R}^T \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{5.17}$$

Giving the texture and vertexes of the polygon, OpenGL is able to select pixels within the polygon, store them and display them on the screen. Figure 5.19 shows an image with building models projected on it and textures that are generated from the image.

Figure 5.19 Model projections on one frame (left), and textures for two model surfaces (right)

5.7.2 Texture generation

The above section showed how to solve the transformation from images to textures. How to choose image I for the texture from a video image sequence is discussed in this section. The orthogonal image is commonly used as the best image for texture mapping from multiple images. For the terrestrial image sequences used in this thesis, one more problem needs to be considered. Due to the limited space between the camera position

and the building facades, some facades cannot be fully seen in any frame. So the first step is deciding how many frames are needed to recover the texture. Since the ground plane and eaves are visible in most frames, only the horizontal coverage for images needs to be checked. For this processing, the viewing direction is assumed to be orthogonal to the building facade, although most of the time this is not possible.

1. Project surface patch boundary points to the visible frame range for the surface patch. If all the points are valid in any one frame, the number (num) of frames to complete the texture is 1, go to step 4.

2. Calculate the horizontal length $(length)$ for the surface patch.

3. Project the middle point of the surface patch to the visible frame range. When the point projection is inside the frame, the distance (d) between the camera and surface patch is calculated. Then the horizontal coverage (c) for one image is computed from the image width (w) and the camera focal length (f) as:

$$\frac{w}{c} = \frac{f}{d} \tag{5.18}$$

The number (num) of frames to complete the texture is decided by the horizontal length $(length)$ and the average coverage:

$$num = \frac{length}{\sum_{i=0}^{n} c_i} \times (n+1) \tag{5.19}$$

4. Divide the surface patch into some parts based on num. For each part, the frame with the greatest angle between the viewing direction and surface normal vector is chosen as the image source for the texture.

If there is no occlusion, this method is simple and direct. However, as seen in Fig. 5.18, the background buildings may be occluded by foreground objects, such as trees, cars or people. These occlusions make the visualization effect very poor. However, automatically removing occlusions from textures and mending the highly occluded areas still present problems, as discussed in section 2.7. Another option for texture generation from video image sequences is to take advantage of background objects, which must be consistent for a long period, and foreground objects, whose projection cover a pixel for less than 50% of the entire sequence length that the pixel is visible. A decision for each pixel from the image sequence is made individually based on the median value. It is therefore possible to separate background buildings from the occlusions. This method is very useful to remove occlusions that cover facades for a short time. However, if the foreground objects are too close to the building facade, highly occluded areas can still not be automatically detected and they will blur the texture.

Figure 5.20 Texture generation results of two methods: texture generated from four images (upper) and texture generated from all the visible frames (lower)

Two texture generation results for the same surface are shown in Fig. 5.20. The top image is an image mosaic result based on four frames. There are some occlusions in the selected frames for texture, which lead to part of the trees and cars being visible in the texture. The bottom image is computed from the respective median value of corresponding pixels, in which occlusions (except for bushes) in front of the building have been removed. However, the bottom image is less sharp than the top one because there is a small depth difference between the wall and the details on it, such as windows and doors.

6 Performance assessment

This chapter provides an assessment of the performance of the building reconstruction method presented in the previous chapters. One of the difficulties in building reconstruction is the successful recovery of geometric building structures with topological relations. In order to solve this problem, knowledge on building structures is integrated into the reconstruction steps. The performance assessment has to check the correctness of each reconstructed building. In general, performance assessment involves evaluating the performance of single modules and the complete system, as well as a qualitative and quantitative assessment of the results. In the following sections, however, only the quality of building models is evaluated.

This chapter is organized as follows: Section 6.1 describes the test data. It was recorded using a hand-held video camera or one mounted on a moving car. Section 6.2 analyses the expected geometric accuracy based on the input data and elaborates on the evaluation rules. The reconstruction results are described and analyzed in section 6.3, which are followed by conclusions in section 6.4.

6.1 Data description

1) Image sequence 1

The video was captured by a Canon camera. A person walked along one side of a street and held the camera so that it was almost facing the building facades. The images have a resolution of 640×480 pixels and a frame rate of 30 frames per second. There are 134 frames in total in this case. This street has a long, connected building and is located in Enschede, the Netherlands. The image sequence only captured some of the building's facades, which will be further referred to as building I (Fig. 6.1).

(a) first frame (b) middle frame (c) last frame

Figure 6.1 Image sequence 1, building I

2) Image sequences 2 and 3

These two videos were captured by a hand-held SONY camera with the same resolution and frame rate as image sequence 1. Some frames are shown in Fig. 6.2. Most of the image sequences used in this thesis were captured by this camera. For image sequence 2, there are 126 frames in total, and for image sequence 3 there are 189 frames. Due to the limited space in this street, the viewing angle with respect to the normal vector of the wall was usually larger than 45 degrees in order to show a wall completely in one frame.

These two buildings II and III are also located in Enschede. Some parts of them have only one storey, while other parts have two storeys.

(a) first frame (b) middle frame (c) last frame

Figure 6.2 Image sequences 2 and 3, buildings II (top) and III (bottom)

3) Image sequence 4

This image sequence was taken by a camera mounted on a trolley and the film was preprocessed before this experiment. Therefore 64 undistorted images with a resolution of 2448×1836 pixels are used. Specific information on this camera is unknown. The stadium, building IV, shown in the images is the Women's Basketball Hall of Fame (WBHOF) in Knoxville, Tennessee, USA. Only the planar parts that are visible in the images are reconstructed (Fig. 6.3).

(a) first frame (b) middle frame (c) last frame

Figure 6.3 Image sequence 4, building IV

4) Image sequences 5 and 6

A person walked along these two buildings, V and VI, and tried to capture all the surfaces that are visible from the ground. Building V in image sequence 5 is the same one as in image sequence 2. These two videos were captured by a Panasonic video camera under its 'progressive scan mode'. The image resolution is 720×576 pixels. Image sequence 5 has 1415 frames and image sequence 6 has 840 frames.

(a) first frame (b) middle frame (c) last frame

Figure 6.4 Image sequences 5 and 6, buildings V (top) and VI (bottom)

5) Image sequences 7, 8 and 9

These three image sequences were recorded to test the method's usability when there are many buildings on a street. For these tests, one person sat in the back of a car holding the Sony camera that was mounted on a tripod to keep it stable. The speed of the car was around 15 km/h for the first two sequences (7and 8) and 40 km/h for the last sequence (9). The viewing angle with respect to the road was between 45-90 degrees.

Figure 6.5 Car trajectory of image sequence 7 (top image:© Blom), some individual frames (bottom)

Car trajectories are shown in Pictometry oblique images. The buildings to be reconstructed are on the car's right-hand side. Image sequence 7 in Fig. 6.5 has 892 frames. This street is in a residential area, so only a few cars and people were visible during the working day. Image sequence 8 in Fig. 6.6 has 1268 frames. There are some occlusions, such as bushes, trees and cars in front of the buildings. The car was in the right-hand lane of the road, so there was a short distance between the camera and building facades. Image sequence 9 has a similar situation as image sequence 8. However, the car was then driving in the left-hand lane and going at a higher speed. More buildings were included and there are 429 frames all together, with motion blur in some frames.

Figure 6.6 Car trajectory of image sequence 8 (top image:© Blom), some individual frames (bottom)

Figure 6.7 Car trajectory of image sequence 9 (top image:© Blom), some individual frames (bottom)

6.2 Assessment aspects

Surface completeness, surface correctness, geometrical accuracy and topological correctness of the reconstructed models were chosen to assess the success of the method. Since image-based modeling systems are always sensitive to the parameters, the parameter settings are introduced first. The parameters used in this thesis can be separated into two different types. One type is related to the image processing and the other type is defined in object space. There are also some parameters that are requested by algorithms, as described in section 2.2.2, such as the Canny detector and Hough transformation. Those parameters are decided by image quality and the objects in the image. Without sufficient experience on setting parameters for those algorithms, the appropriate parameters are usually decided based on several experiments. It is similar case about thresholds related to the image processing in this work. As mentioned before, building detection was not incorporated in this research, and the reconstruction steps are only applied to data from which enough features could be extracted to support the existence of buildings. It also means the accuracy of the reconstructed models depends on the accuracy that can be achieved by the input data.

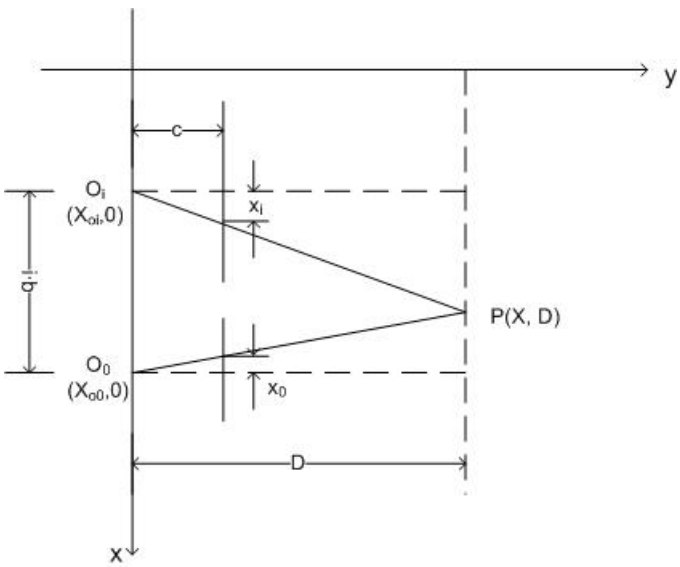


Figure 6.8 The normal case for terrestrial video image sequences

In the "normal case", the camera is moving at a constant speed along a straight line and recording perpendicular to its trajectory. Therefore, the distance between two adjacent camera positions can be assumed to include the same base length b. A point $P(x,y,z)$ is observed in K frames with camera positions $O_i(x_{oi},0), i=0,\cdots,K-1$, and its projections in the images' x coordinates are $x_i, i=0,\cdots,K-1$ with a standard deviation σ_x. With a constant focal length c, Fig. 6.8 shows two photographs that are captured at camera positions O_0 and $O_i(i \neq 0)$ of the point P, therefore the distance between O_0

and O_i is $i \cdot b$. For simplicity, the point and camera lie in the same *XY*-plane; the camera's y coordinates are 0 and the point's y coordinate is D . This condition is very difficult to achieve for terrestrial photographs but can be used to estimate a rough accuracy. According to the theory of error propagation for photogrammetry and neglecting the errors in *c*, *b* and the camera orientation, the standard deviations in depth and in the *XZ*-plane coordinates by (Förstner, 1998):

$$\sigma_D = \frac{D^2}{c \cdot b} \cdot \frac{\sqrt{12} \cdot \sigma_x}{\sqrt{K(K^2 - 1)}} \tag{6.1}$$

$$\sigma_{X,Z} = \frac{D}{c} \cdot \frac{\sigma_x}{\sqrt{K}} \tag{6.2}$$

Taking the Sony camera used most in this thesis work as an example:
Sensor size 7.18 mm×5.32 mm
Focal length 21 mm
Resolution 640×480 pixels
Frame rate 30 p
Walk speed 1 m/s

$$pixelsize = \frac{7.18mm}{640} = \frac{5.32mm}{480} \approx 11\mu m$$

$$\sigma_x \approx pixelsize = 11\mu m$$

$$b = \frac{1m}{30} \approx 3.3cm$$

Table 6.1 Theoretical accuracy $(\sigma_{X,Y}, \sigma_D)$ of eight sequences

Image sequence	$\sigma_x(\mu m)$	$c(mm)$	$b(cm)$	K	$D(m)$	$\sigma_D(mm)$	$\sigma_{X,Z}(mm)$
1	11	20	3.3	80	10	8.1	1.0
2	11	21	3.3	100	15	12.4	1.0
3	11	21	3.3	80	12	11.1	0.7
5	9.5	18.7	3.3	100	12	7.7	0.6
6	9.5	18.3	3.3	100	6	2.0	0.3
7	11	21	13.9	80	11	2.2	0.6
8	11	21	13.9	60	10	2.8	0.7
9	11	21	37	35	11	2.9	1.0

In order to have building eaves visible in the images, the camera positions cannot be too close to the building. However, limited space in most streets leads to the distance between the camera and building walls was mostly 10–15 m. In Table 6.1, the root mean square errors for all sequences, except sequence 4 (whose camera information is unknown), are presented based on frame numbers (K) and approximate distances (D) between the camera and walls. K is an approximate average value of frame range in

which object points can be visible. For short image sequences captured by a walking person, the distances between the first and last camera positions are short, leading to a low depth accuracy. The theoretical values show a very high accuracy can be reached for input image sequences. However, such accuracy is difficult to achieve in reconstructed surfaces because there are camera orientation errors and errors in grouping sparse features. According to the accuracy requirements defined in CityGML (Open Geospatial Consortium, 2008), the absolute 3D point accuracy has to be lower than 0.2 m for LOD4. For other level of details, such as LOD2 and LOD3, which were the aim of this research, a lower accuracy is acceptable. To enable the possibility of reconstructed models being used for a higher level of detail, 0.2 m was chosen as the default value of the maximum distance of a point to the plane and the minimal threshold of parallel edges. Thus, facade details, such as windows and doors cannot be separated from the main facade if only according to the geometry. The geometric expectation of reconstructed models is within the distance tolerance for points to plane. The assessment is applied to the main building facades.

As the results are based on a coordinate system computed from the first few frames, without defining the coordinate up-to-scale as world coordinate, the absolute value cannot be directly compared with measurements. Although, during the preprocessing step, it is possible to define a coordinate up-to-scale by known distances between feature points in the world coordinate system or by survey points in the same world coordinate system. It is not always possible to get these accurate reference data, although some distances can be assigned by experience in order to set an almost similar scale. The model geometry assessment mainly depends on the following two internal checks:

- Angle relations of surfaces in reality should be the same as in the model (parallel, orthogonal, etc.)
- Distance relations in reality should be the same as in the model (constant distance, different floor levels, etc.).

The completeness can be determined by comparing the number of reconstructed surface patches with the visible ones from image sequences. The semantic meaning of surface patches can be identified by the corresponding label. Since topological relationships between surface patches have already been used in the reconstruction, the reconstructed models can confirm their relations. Textures in this thesis were used to visually check the correctness of models and no quality evaluation was done for them.

6.3 Results and evaluation

Figure 6.9 shows the building reconstruction result of image sequence 1. Since the video image sequence only shows parts of the facades along the street, the building model is not complete. Four surface patches were observed and generated from the image sequence. Together with two self-occluding surfaces (a wall and a roof) that were recovered from the hypothesis, six surface patches were reconstructed with the correct topological relation. The texture of the right facade includes some sky projection due to the building's design and that fact that this facade was generalized into a rectangle shape.

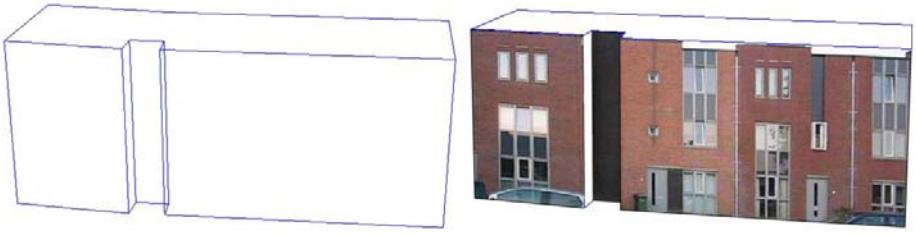


Figure 6.9 Reconstruction result of building I in image sequence 1, without texture (left), and with texture (right)

The reconstruction result of image sequence 2 is shown in Fig. 6.10. Only six surface patches of this building were visible in the images and they were reconstructed with correct topological relations. Part of surface patch 6 was always occluded by other surfaces, which led to texture errors in the textured model.

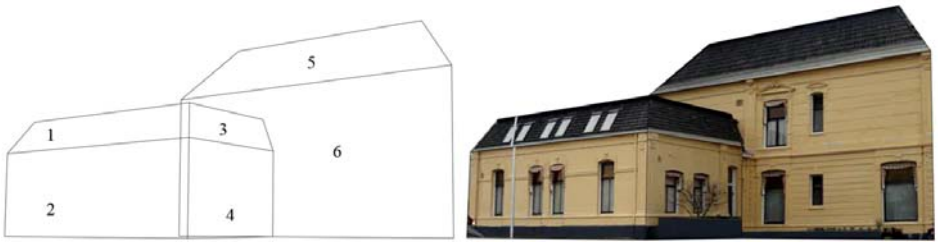


Figure 6.10 Reconstruction result of building II in image sequence 2, without texture (left), and with texture (right)

For the building in image sequence 3, three parts could be observed from the images. One part only had one storey. Another part had two storeys and one big extrusion was attached to it. Some self-occluded surfaces were reconstructed based on general building structure hypotheses (Fig. 6.11). The same as building II, self-occlusion resulted in texture errors in surface patches 1, 2 and 4.

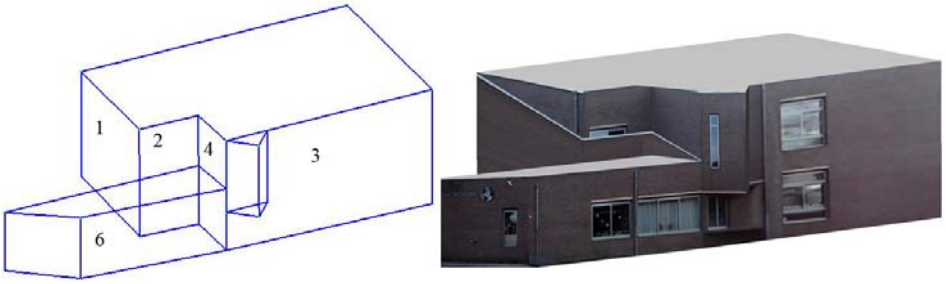


Figure 6.11 Reconstruction result of building III in image sequence 3, without texture (left), and with texture (right)

Some surface patches of the stadium in image sequence 4 were almost on the same plane and only a few reliable features were extracted from the roof and left side of the building. To avoid the extension of surface patches to areas belonging to other patches, the defined feature grouping rules played an important role. The recovered structure

presented the planar part of the stadium and it satisfied the requirements of LOD2. Meanwhile, some surface patches were lost due to few features being extracted in those areas, such as intrusion parts (there are two missing planes in the red circle in Fig. 6.12).

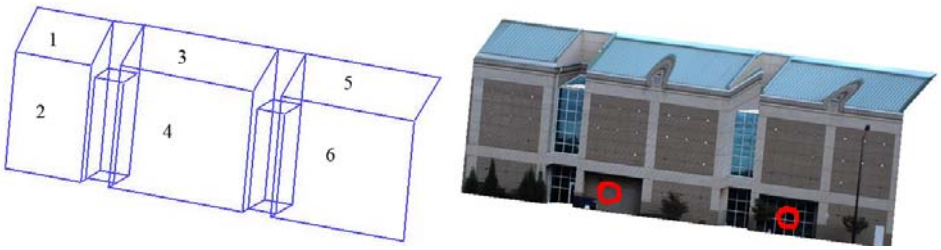


Figure 6.12 Reconstruction result of building IV in image sequence 4, without texture (left), and with texture (right). Red circles (right) indicate planes missing in the model

The relations between surface patches are used to evaluate the results. According to the preferential knowledge, as defined in section 3.2, a building ground plane is mostly right-angled. The angles are computed for extracted surface patches that are visible in the image sequence and which could stratify this rule. The angle between potential parallel surface patches is also computed, as shown in Table 6.2. For buildings II and III, the deviations are less than 5 degrees. For building IV, three similar surfaces 2, 4 and 6 show their small differences in normal vectors. Since the slope between an oblique roof and its intersecting wall can be similar within the same building, the angles between surfaces 1 and 2, 3 and 4, and 5 and 6 in buildings II and IV are considered. The results confirm the similar angles between building planes with the same relation.

Table 6.2 Angles (degree) between surface patches of buildings II, III and IV

Building no.	1 and 2	3 and 4	5 and 6	2 and 4	4 and 6	2 and 6
II	51.34	48.67	52.75	89.35	91.13	0.58
III	94.22	93.17	–	92.29	90.59	1.71
IV	60.44	62.25	62.25	0.50	0.19	0.67

Although it is unreliable to compare absolute values directly when the scale of the local coordinate system is not the same as that of the world coordinate system, the differences between reconstructed models and reference data, or within reconstructed models, are useful to indicate surface relations and evaluate the results. Firstly, parameters have to be described, because they affect the success and the level of detail of the results. For parameter settings, the approximate scale difference between the local coordinate system and the world coordinate system is considered. As mentioned above, such a difference can be removed by some reference distance measurements. For buildings I and II, some estimations of general knowledge, e.g. common window heights, were used. No length reference was applied to building III during preprocessing and the scale seemed much smaller than general estimations, so thresholds related to distance were reduced to half of the default setting. For building IV, the image quality was much better than the other three cases and the point density was also higher. In the point segmentation step, 0.1 m was selected for the maximum distance between point and

plane and there was no reference data for the image sequence. This value was also chosen for the maximum distance between point and surface patch in the reconstruction step. Other parameters were still default values.

The angle between surfaces 3 and 6 of building III was 1.58 degrees and angles between surfaces 2, 4 and 6 of building IV were less then 1 degree. The distances between these almost parallel surfaces ware computed by the average distances from the boundary points of one surface to the other surface. The absolute distances between these surfaces are listed in Table 6.3, which can be interpreted as them being coplanar within the distance threshold. Although these differences were larger than the theoretical expectation, i.e. $\sqrt{2}$ times the σ_D from Table 6.1, they were smaller than the distance threshold of points to plane. This is because the theoretical expectation does not include errors of camera orientation or errors of feature grouping. Furthermore, the extracted features were not dense enough to identify details of the main facades. The accuracy achieved, however, satisfies the aim of the research and was within the actual geometric expectation.

Table 6.3 Distances (m) between surface patches of buildings III and IV

Building no.	3 and 6	2 and 4	4 and 6	2 and 6
III	0.14	–	–	–
IV	–	0.06	0.04	0.02

There are available airborne laser scanning data with a point density of 20 pts/m^2 over the city of Enschede, in which buildings II and III are located. Although facades cannot be fully seen, this data can provide a reference for roofs. Building V presents the same building as building II, but with more facades. The heights of different floors were measured from laser data and compared with the reconstructed results (Table 6.4). The two model reconstruction results of the building have similar ratios to the laser scanning data and the absolute values for building (3) confirmed that the scale of its local coordinate system was about half that of the world coordinate system.

Table 6.4 Height values and ratios of surface patches of buildings II, III and V

Building no.	2 (laser)	6 (laser)	2/6 (laser)	2 (result)	6 (result)	2/6 (result)
II	5.16	10.03	0.51	4.91	9.47	0.52
III	7.77	4.35	1.79	3.29	1.84	1.79
V	–	–	–	5.59	11.04	0.51

For building IV, there was no reference data. However, from images in Fig. 6.3 and 6.12, the intersecting edges between surface patches 3 and 4, and 5 and 6 seemed to have a similar length. In a local coordinate system, the length ratio of these two edges was 1.01, which is quite a reasonable difference.

Image sequences 5 and 6 were used to test how well the proposed methods could reconstruct complete building models. The default parameters were used for image

sequence 5. The shortcoming of this sequence was that trees occluded some building facades. With the accumulated error, the orthogonal relation between four walls of the large part could not be confirmed (Table 6.5). For the small part, a box hypothesis for the walls was made from three visible ones. However, the invisible surface 13 was not in fact in the same plane as surface 6 (Fig. 6.13). The texture model is shown in an OpenGL mode so that both the front and back surfaces are visible.

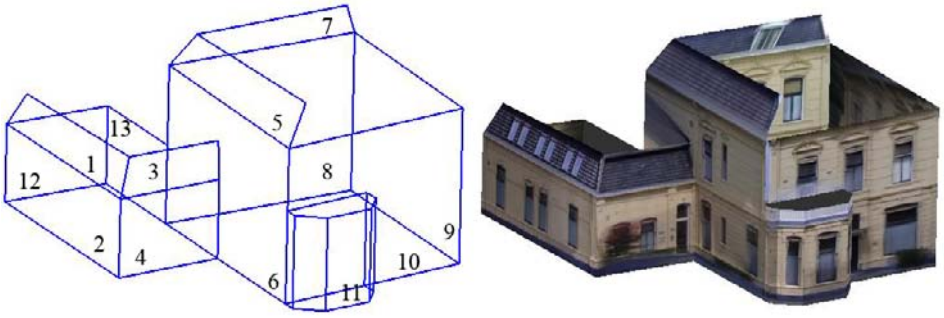


Figure 6.13 Reconstruction result of building V in image sequence 5, without texture (left), and with texture (right)

Table 6.5 Angles (degree) between surface patches of buildings V and VI

Building no.	1 & 2	3 & 4	5 & 6	7 & 8	2 & 4	4 & 6	6 & 8	2 & 8
V	51.09	46.75	51.59	50.70	92.04	92.04	90.32	90.32
VI	46.73	51.38	42.59	50.09	90.85	89.62	87.35	92.18

	2 & 12	6 & 10	8 & 9	9 & 10	10 & 11
V	90.32	95.81	89.88	83.98	2.1

Building VI in image sequence 6 has a very simple structure with many windows located in the walls. Since many feature points and edges were extracted from the windows, a larger threshold (0.3 m) was chosen for the maximum distance between a point and the plane to merge details into the walls. Extracted walls confirmed it could be shaped as a box (Table 6.5). However, it was not easy to observe roof parts from ground level, so the roof surfaces were not accurate, which led to the angles between the walls and roofs being quite different and the sky being projected to part of the roofs. A model without curved surfaces is shown in Fig. 6.14.

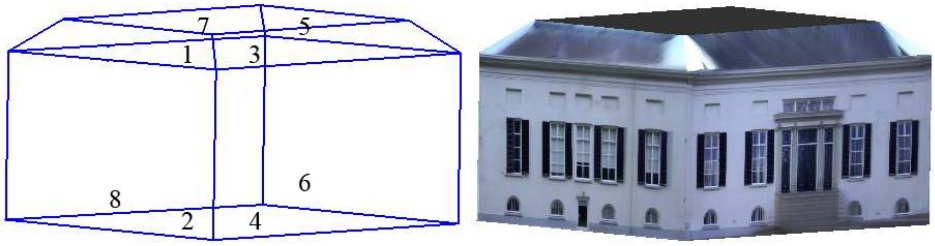


Figure 6.14 Reconstruction result of building VI in image sequence 6, without texture (left), and with texture (right)

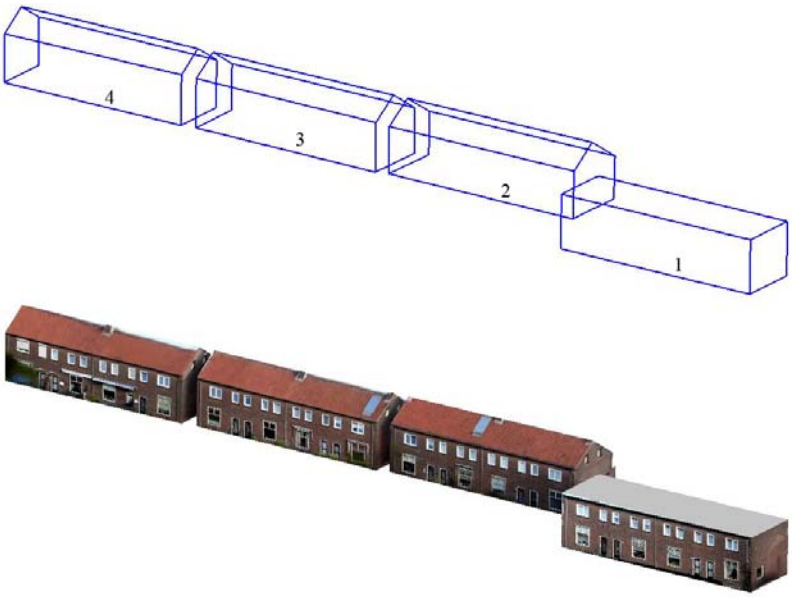


Figure 6.15 Reconstruction results of buildings in image sequence 7, without texture (top), and with texture (bottom)

Figures 6.15, 6.16 and 6.17 show reconstruction results for more than one building on the streets. The buildings have a similar complexity and the image resolutions are the same. However, due to the speed of the car and conditions in front of the buildings, the results were different. The street in image sequence 7 is in a residential area, only a few cars and people were therefore evident during working hours. With the default thresholds, the surface patch generation result was quite good. Only the roof of the first building was not reconstructed successfully because it was not visible in the image sequence. The sizes of these four buildings seem the same as shown in the image. The length and height of the front facades of these buildings (Fig. 615) were measured based on a local coordinate system and are listed in Table 6.6. As mentioned above, the roof of the first building was invisible and without the effect of an eave, so surface patch 1 was higher than the other three surfaces. Angles and distances between front facades 2, 3 and 4 are shown in Table 6.7, and confirmed that they are coplanar surfaces. Although

the differences between length, height and depth were larger than the theoretical expectation, the differences between height and depth were within the distance threshold, which was within the actual geometric expectation, while the quality of length needs to be improved. Since one frame could not cover the whole facade length, more errors were accumulated in the length than in the height and depth.

Table 6.6 Length and height of front facades of buildings in image sequence 7

Facade no.	1	2	3	4
Length (m)	26.08	26.01	24.98	24.53
Height (m)	6.68	6.22	6.18	6.17

Table 6.7 Angles and distance between surfaces

	2&3	3&4	2&4
Angle (degree)	0.10	0.55	0.46
Distance (m)	0.09	0.14	0.12

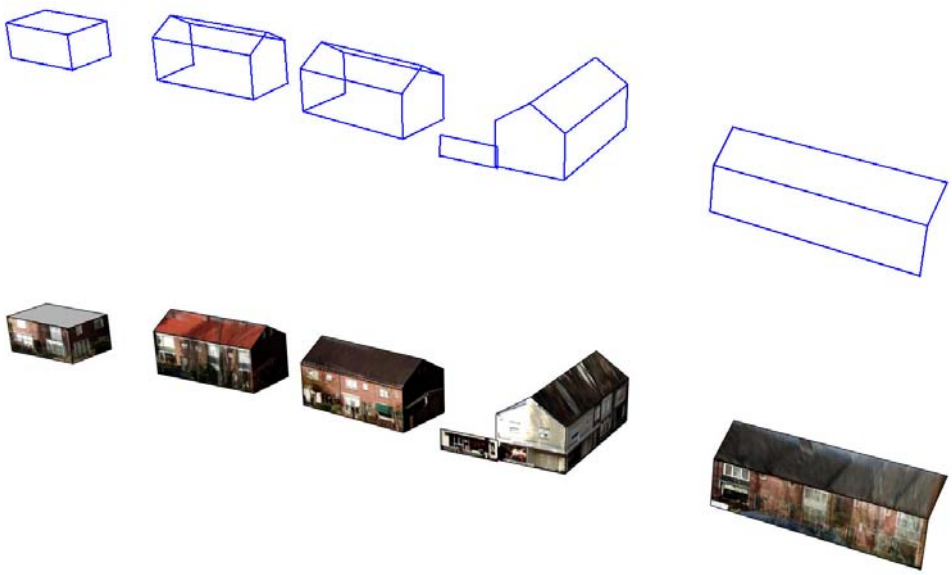


Figure 6.16 Reconstruction results of buildings in image sequence 8, without texture (top), and with texture (bottom)

Some occlusions, such as bushes, trees and cars are seen in front of the buildings in sequence 8. Driving at a slow speed, the car had to be in the right-hand lane, which meant there was only a short distance between the camera and building facades. The poor results for feature extraction could not be avoided because images were only captured from one direction. Larger distance thresholds, which reduced the possibility of separating different objects, were chosen. Because of these weaknesses, there were

some mistakes in the reconstructed result. The facades of the two nearby buildings on the right-hand side of the image were merged together and their textures were blurred (Fig. 6.16). The other four buildings could be identified separately based on the large gaps between them. Parts of the facades of the second building and the roof of the last building were lost.

The situation of the street in image sequence 9 was similar to that in image sequence 8. Driving at a higher speed, the car was in the left-hand lane. The distances between the camera positions and building facades were greater and made a large difference; each facade was visible in approximately 35 frames and the viewing angles with respect to the driving direction were almost constant at 45 degrees. Some facades were therefore missed or only partly reconstructed (as shown in Fig. 6.17) because they were invisible in the images or too few features were extracted. There were ten buildings visible in this image sequence: two of them that contained facades 1 and 6 could not be recovered. Although the other eight buildings had different roof structures and extrusions, such as chimneys, dormer windows, and overhangs, they were all two-storey buildings (Fig. 6.18). Their generalized models could belong to the same type. However, the differences in roof structures still affected the results, e.g. buildings 8 and 9 had the third roof structure, as shown in Fig. 6.18. The building top was invisible from ground-level, so that slanted roof planes did not reach the top of the generalized side facades. The lengths and heights of the front facades of these buildings were measured in the local coordinate system and are listed in Table 6.8. The complex environment on the streets and the motion blur in some frames (due to the high speed) reduced the numbers of frames in which each facade was visible and thus increased errors in feature extraction and camera orientation. The quality of the reconstructed models was, therefore, worse than the theoretical expectation.

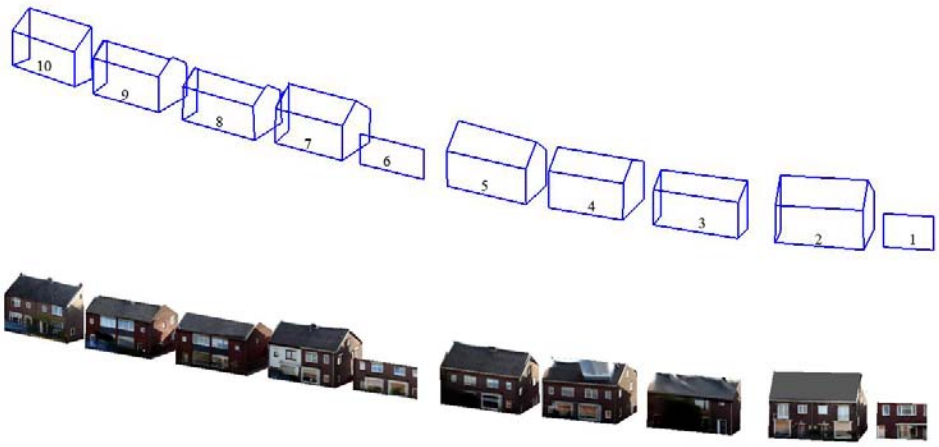


Figure 6.17 Reconstruction results of buildings in image sequence 9, without texture (top), and with texture (bottom)

Figure 6.18 Differences in roof structures in image sequence 9

Table 6.8 Lengths and heights of building front facades in image sequence 9

Facade no.	2	3	4	5	7	8	9	10
Length (m)	13.12	13.15	12.54	13.03	12.25	13.23	12.44	12.37
Height (m)	6.46	6.21	6.41	6.12	6.02	5.86	6.23	6.09

The completeness evaluations for the buildings in all the image sequences are given in Table 6.9. 'Observed surface patches' indicates the number of surfaces that were visible in the scene and that could have been reconstructed in theory. 'Reconstructed surface patches' is the number of surfaces determined in the reconstructed model. 'Self-occluded surface patches' indicates the number of surfaces that were reconstructed but that were invisible in the image sequence. "Wrong surface patches' is the number of surfaces with a wrong plane position. 'Missed surface patches' is the number of surfaces that were visible but not reconstructed. The relation between them can be represented by: $R = O + S - M$. The reasons for wrong and missing surfaces have been given above, but as a conclusion the average completeness was 94% ($\frac{R}{R+M}$) and the correctness was 97% ($\frac{R-W}{R}$).

Table 6.9 Completeness evaluations for the reconstructed buildings in all the image sequences

Building in image sequence	Observed surface patches (O)	Reconstructed surface patches (R)	Self-occluded surface patches (S)	Wrong surface patches (W)	Missed surface patches (M)
1	4	6	2	0	0
2	6	6	0	0	0
3	7	12	5	0	0
4	14	16	4	0	2
5	19	18	2	1	3
6	6	8	2	0	0
7	14	18	5	1	1
8	18	19	4	3	3
9	29	32	6	0	3

The test environment was as follows: a CPU Intel Core2 Duo T9600 at 2.80 GHz, 3 GB of internal memory, and Windows XP Professional operating system. The time spent on each step is shown in Table 6.10 for all the image sequences. Preprocessing includes

manually estimating the distortion, feature-point tracking and camera orientation with Boujou. For long sequences, automatic feature tracking may not find enough feature tracks or cause some errors, which increases the time for computing relative camera positions for some conjunctive frames. One optional step after preprocessing is to transform from the local coordinate system to the world coordinate system by making use of some reference distance measurements. This takes about 3 minutes for manual work and is not included below. The main disadvantage of this table is that it does not include the time for the operator to analyze the results for each step and it shows the time when appropriate parameters had been chosen for the algorithms. The parameters for image processing depended on the operator's experience and the data properties. Therefore, preprocessing and edge extraction may cost more time than that given below, but when acceptable results are obtained, the parameters for object space for further reconstruction steps are easy to choose. The time for generating surface patches is related to the number of surface patches, while the time for reconstructing the model is related to the number of buildings and their complexity.

Table 6.10 Reconstruction speeds for all the image sequences

	1	2	3	4	5	6	7	8	9
Number of frames	134	126	189	64	1415	840	892	1268	429
Number of buildings	1	1	1	1	1	1	4	5	10
Observed surface patches	4	6	7	14	19	6	14	18	29
	Time in minutes								
Preprocessing	1	1	2	1	30	13	100	73	32
3D edge extraction	5	5	6	3	8	4	6	8	8
Surface patch generation	1	1	1	2	2	1	2	2	2
Model reconstruction	1	1	1	1	2	1	2	1	1
Texture mapping	1	1	1	1	2	2	2	2	1

To analyze the robustness of the proposed reconstruction method, more tests on the parameter settings were applied to the image sequences. However, parameters for image processing and point-cloud segmentation will not be discussed. The parameters used in the reconstruction steps and introduced in earlier sections are the maximum distance between the intersection point and edge (d_1), the minimal distance between two parallel edges (d_2), and the maximum distance between a point and plane (d_3). To keep the same distance tolerance of point-to-plane and edge-to-plane, d_1 and d_2 were set at the same value during the experiments. Table 6.11 shows a comparison between the adapted parameters and those used in the above experiments for image sequences 6 and 7 from five aspects: the distance difference between models (A), the angle difference between models (B), topology correctness between surfaces (C), surface completeness (D), and surface correctness (E). The first two aspects were compared when the model reconstruction results were correct. The first rows of each image sequence show the results that are presented in Figs. 6.14 and 6.15, which are acceptable results after human interpretation. They were used as a reference for comparing the other parameter settings. Hence, no distance and angle differences are given in the first rows. Firstly, d_3 kept the same value as the reference experiment, $d_1(d_2)$ took smaller and then larger thresholds. After that, $d_1(d_2)$ stayed the same as the reference, and d_3 was changed.

Table 6.11 Comparison of the different parameters used in the reconstruction steps, – indicates results based on parameters in that row were used as references, and × indicates results based on parameters in the row were not compared with references.

	d_1, d_2	d_3	A	B	C	D	E
Image sequence 6	0.3	1	–	–	Correct	100%	100%
	0.2	1	×	×	Wrong	81%	56%
	0.4	1	0.09	0.2	Correct	100%	100%
	0.3	0.5	0	0	Correct	100%	100%
	0.3	2	0	0	Correct	100%	100%
Image sequence 7	0.2	1	–	–	Correct	95%	94%
	0.1	1	0.01	0.1	Correct	95%	94%
	0.05	1	×	×	Wrong	95%	52%
	0.5	1	×	×	Wrong	94%	48%
	0.2	2	0.12	0.3	Partly correct	53%	87%
	0.2	0.5	0.01	0.2	Partly correct	94%	94%

For image sequence 6, if the default or even smaller thresholds of $d_1(d_2)$ were used, some small surfaces that were caused by features located on the windows were generated and some of them were wrong. These small surfaces increased the difficulty of identifying the main walls and the model reconstruction result was wrong. However, for such a simple structure as in this sequence, large thresholds of $d_1(d_2)$ could be chosen and did not change the reconstruction results. As shown in table 6.11, d_3 has little effect on the results. A value (0.1 m) of $d_1(d_2)$ was chosen for image sequence 7. Using the new threshold, the reconstruction result was almost the same as the original one, with a very small difference between the plane parameters. If a smaller value than 0.1 m for $d_1(d_2)$ was chosen, the surface patches begun to break into several parts, which led to a wrong model reconstruction. If large values for $d_1(d_2)$ were chosen, features on objects in front of the buildings were grouped with some points on the walls to form small surfaces, which also leads to wrong reconstruction. According to the defined concept, completeness for these cases was still high, but correctness was low. Since three walls in this case were coplanar, a large threshold for d_3 led to them being grouped into one surface. Three buildings were therefore generalized as one big building, and the geometric differences were small with a low completeness. A small threshold for d_3 could lead to the coverage of surfaces being very small, for example the wall should connect with surface patch 1 in Fig. 6.15 and would have to be extended too far if they formed a local model, which would then be marked as unreliable in the proposed method. Thus, they were not connected in the reconstructed model.

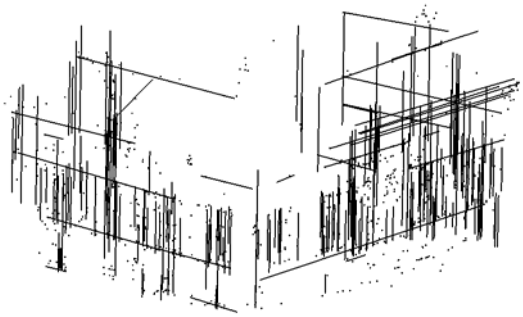


Figure 6.19 Feature extraction result for image sequence 6

For data-driven methods, extracted features affect the reconstruction results. First, they affect surface patch generation, and models cannot be recovered without surface patches. The following discussions are based on at least 80% of surface patches still being recovered if parameters are changed. Feature extraction results show that the distribution of features is related to building textures, and some important features, such as corners and boundary edges, may not be extracted. Figure 6.19 shows the feature extraction result for image sequence 6. To test the effect of features on the proposed method, parameters of feature extraction were first changed. To keep the correctness of feature extraction results, parameters for feature extraction cannot have a very low sensitivity to noise. The number of extracted features cannot be significantly increased and the surface patch generation result is almost the same, with a small difference between plane parameters. If the parameters are highly sensitive to noise, fewer features are extracted. Since the edge extraction is based on reliable feature points, fewer feature points may lead to fewer edges or a failure to group points from point segmentation. Fewer edges cause a failure to generate surface patches or reduce the coverage of surface patches, especially when only a few features for a particular surface patch were extracted. The location of removed edges has a different effect on the results. To determine this, one edge of each surface of building (6) was manually removed from the feature extraction result. Since the structure of building (6) was very simple and many edges can be extracted from the walls, removing an edge located on the walls, even along a surface boundary, does not, change the result of the model reconstruction. For roofs, however, removing an edge can lead to missing a roof surface. If fewer than three roof surfaces are generated, the roof part cannot be connected.

By analyzing all the experimental results, the following conclusions can be drawn:

- Redundant images play an important role in the feature extraction.
- Because of the low resolution of images and the homogeneous color of the building facades, only a few feature points and edges were extracted from the actual building edge areas. The surface boundaries that were obtained from two intersecting neighboring surface patches were more accurate than those based only on feature grouping.
- The reconstruction method can recover some self-occluded surfaces based on general building structure hypotheses. However, it still depended on surface patches that were grouped from features.

- Default parameter values are assumed to be valid in most cases if the scale of the local coordinate system is similar to the world coordinate system.
- For extracted surface patches, the proposed reconstruction method is successful in building up topological relations between surface patches that belong to the same building.
- Since errors are accumulated during all the steps, from the beginning of feature tracking to the final model reconstruction, for long image sequences, the accuracy of reconstructed models cannot satisfy those applications that require a high geometric accuracy. However, they can satisfy the accuracy requirement of LOD3, as defined in CityGML, and of the standard model defined in the "Chinese technical specifications for three-dimensional city modeling (CJJ/T157-2010)".

7 Conclusions and recommendations

A method for the automated reconstruction of building facades from a terrestrial video image sequence has been presented in this thesis. Various problems have been discussed, including the extraction of main surface patches, setting up topological relationships between them, and the recovery of the building shape structures. The conclusions on each step have been given in the previous chapters. This chapter concludes with an overview of the approaches in section 7.1 and a description of the perspectives for further research in section 7.2.

7.1 Conclusions

The work presented in this thesis still has some limitations. The first is that it does not include a building detection step, which means buildings must be visible in the image sequences. The image sequences therefore need to be planned before image capturing begins. The second limitation is that the proposed reconstruction method is based on surface patches that are grouped from extracted feature points and edges. If there are not enough features to support the existence of a surface patch, it cannot be extracted. Missing one surface patch is not a great problem, but if only one facade of a building is extracted, it is impossible to reconstruct the whole building. Apart from surfaces that are invisible in the images, there are many reasons why surfaces may not be recovered. Feature points and edges correspond to pixels with salient intensity changes in images. When a surface has a homogenous color and/or texture, or if it is in shadow, the features are difficult to detect. Occlusions and the quality of image have a significant influence on the feature tracking process. For sparse 3D features, different parameters may lead to totally different grouping results. Parameters have to be set and checked by humans to obtain an appropriate result. Due to the way in which video images are captured, some areas are only visible in the first and last few frames, while fewer reliable features are extracted from areas based on a short baseline. Such factors affect the extraction of surface patch boundaries.

For extracted but sparse 3D points and edges, the proposed method sets up rules and processing steps to group them together reasonably under the guidance of prior building structure knowledge. Then the surface patch outline helps to restrict planes in the region corresponding to the actual case and makes it easier to connect adjacent surface patches later. Only the geometric information conveyed by the 3D points and 3D edges is used for generating the surface patches. Image information, especially on the intensity and texture, is used in the verification step when new surface patch hypotheses are proposed for areas where few features were extracted.

The proposed building reconstruction method is a hybrid, data-driven and model-driven strategy that is used to connect neighboring surface patches and to reconstruct a building model. The results show that the method correctly sets up topological relationships between the surface patches generated and also establishes a reasonable structure in some areas with occlusions. To integrate prior building knowledge does not

complicate the processing, but rather it simplifies the reconstruction process. The method reconstructs building models from features first, so it does not restrict us to basic structures. Complex buildings can also be identified when there are observations that disagree with the preferential knowledge. On the other hand, the method also depends on the quality of the results of feature extraction, which affects the threshold setting. For cameras with a normal viewing angle, the distance between the camera and buildings has to be large enough to allow facades and at least part of their boundary is visible. The proposed building models are based on the predefined knowledge, but some buildings may not fully satisfy such constraints. Reconstructed building models are presented in a generalized way and the locations of some intersecting edges do not necessarily correspond to actual boundary edges. Accordingly, the accuracy of the surface patches and outline needs to be improved.

7.2 Recommendations

Currently, the feature tracking and camera position recovery are done by commercial software. This still presents some problems when dealing with long sequences acquired from streets. Feature tracking is based on the hypothesis that a feature point will continue to exist in subsequent frames but with a small difference. There are always some frames that cause problems in long image sequences, such that automatic feature tracking cannot find enough feature tracks to compute relative camera positions for adjacent frames. Thus, no smooth camera trajectory for the whole sequence can be obtained and the camera trajectory will break up into several parts with errors. In the meantime, non-consecutive feature tracking and connecting image sequences cannot be solved automatically. These problems have to be considered in the future.

There are many high-rise buildings in large cities nowadays. Due to the limited camera viewing angles and the distances between camera positions and building facades, one camera can only cover a certain size of surface in a frame. Multiple image sequences have been used in some projects to recover dense point clouds with a high accuracy (Zhang et al., 2007; Pollefeys et al., 2008). To increase the possibility and flexibility of the algorithm for high-rise buildings, several cameras are recommended to be used together.

Some extracted features can indicate the location of small structures, such as doors, windows and chimneys. Using the thresholds, some of them are grouped to the main surfaces and some of them are identified as noise. Sometimes even small surface patches can be grouped from them. However these small patches have not been used in the proposed method. If the geometric quality of building models is better, these features become important for recovering small structures and, in this way, reconstructed building models can achieve a higher level of detail.

In the model refinement step, buildings with a rectangular ground plane and roof overhangs are considered. However, there are more structures that could be added to a generalized model, such as windows, doors and chimneys. The geometrical and topological model information, as well as the theoretically available accuracy, could be

integrated into further refinement steps as internal constraints and to trigger the thresholds during the model reconstruction.

Default parameter values are assumed to be valid in most cases, but in specific situations, they may have to be changed. These changes depend on the scale of the local coordinate system and feature extraction results. It is of great interest to analyze the possibilities for automatically determining the correct parameter values, and particularly meaningful when dealing with long sequences that contain buildings with different sizes, geometric complexities or textural complexities.

The knowledge pool on building structures should be further developed to permit the reconstruction of more types of buildings or of complex buildings. For example, surface patches on different storey of a building have different relations. The improvements to the knowledge pool should focus on two aspects. Firstly, complex spatial relations should be considered, e.g. relations concerning curved surfaces. Secondly, a self-learning system should be built up by the statistical analysis of training samples.

Making use of mobile mapping system has been interested in recent years for many people with a photogrammetry or computer vision background. It would be interesting to fuse terrestrial laser scanning data with video image sequences and use them in the applications that require a high geometric accuracy and realistic textures.

Appendix

Term definition

- Feature

Points, edges and regions refer to 2D features in the images and 3D features in the world.

- Object

Object is a thing, that is tangible and within the grasp of senses, such as buildings, bodies, face, etc.

- 3D model/building model

The way to present/describe the geometry of objects (Mortenson, 1997), such as TIN, CSG. 'Building model' is used when the target object of 3D model is a building.

- Detection

"Feature detection" refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. "Object detection" means that objects are found based on simpler features and camera models (Mayer, 1999).

- Reconstruction

"Reconstruction" means to find a valid description for a particular object. For a highly accurate "reconstruction," knowledge about the object's geometry and especially its topology is assumed to be given (Mayer, 1999).

- Extraction

"Extraction" is a process of retrieving data out of data source. It can be used for features and objects. When it is used for 2D features, it means feature detection and 2D coordinates recovering. When the target is 3D features, it contains 3D coordinate recovering from 2D features. When it is used for objects, it includes both detection and reconstruction.

- 3D object modelling

The complete process starts from data acquisition and ends with a 3D virtual model visually interactive on a computer (Remondino and El-Hakim, 2006). Sometimes there is an object detection step before object reconstruction. However, it is not always required or reconstruction is only applied when objects can be found. It can also be divided into range-based and image-based according to the difference of data.

- Image-based (3D object) modelling

Image based modeling is a widely used method for geometric surfaces of architectural objects, which uses 2D image measurements (correspondences) to recover 3D object information through a mathematical model (Remondino and El-Hakim, 2006). When the target object is building, image-based modelling and (3D) building reconstruction with images are the same. These methods can be automatic or semi-automatic. They also can be categorized into data-based or model-based.

- Image-based rendering

A collecting of sample images are used to render novel views (Shum and Kang, 2000). It does not include the generation of a geometric 3D model.

Bibliography

Ackermann, F. and Krzystek, P., 1995. New Investigations into the Technical Performance of Automatic DEM Generation. Proceedings of ASPRS/ACSM Annual Convention 2, 488-500.

Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J. and Ogden, J.M., 1984. Pyramid Method in Image Process. RCA Engineer 29 (6), 33-41.

Akbarzadeh, A., Frahm, J.-M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H., Nistér, D. and Pollefeys, M., 2006. Towards Urban 3D Reconstruction from Video, 3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT06), Chapel Hill, North Carolina, USA, pp. 1-8.

Ameri, B. and Fritsch, D., 2000. Automatic 3D building reconstruction using plane roof structures, ASPRS Congress, Washington, DC, pp. 22-26.

Amirshahi, H., Kondo, S., Ito, K. and Aoki, T., 2008. An Image Completion Algorithm Using Occlusion-Free Images from Internet Photo Sharing Sites. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 91 (10), 2918-2927.

Ando, S., 2000. Image Field Categorization and Edge/Corner Detection from Gradient Covariance. IEEE Transactions on Pattern Analysis Machine Intelligence 22 (2), 179-190.

Armstrong, M. and Zisserman, A., 1995. Robust Object Tracking, 2nd Asian Conference on Computer Vision. Springer, pp. 58-62.

Atiquzzaman, M., 1992. Multiresolution Hough Transform-An Efficient Method of Detecting Patterns in Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (11), 1090-1095.

Autodesk, 2011a. Homepage Autodesk 3ds Max software. http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13567410 (accessed 2011.3.11).

Autodesk, 2011b. Homepage Autodesk ImageModeler software. http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=11390028 (accessed 2011.3.11).

Baillard, C., Schmid, C., Zisserman, A. and Fitzgibbon, A., 1999. Automatic Line Matching and 3D Reconstruction of Buildings from Multiple Views. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 32 (Part3-2W5), 69-80.

Baillard, C. and Zisserman, A., 1999. Automatic Reconstruction of Piecewise Planar Models from Multiple Views, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), pp. 559-565.

Baltsavias, E.P., 2004. Object Extraction and Revision by Image Analysis Using Existing Geodata and Knowledge: Ccurrent Status and Steps towards Operational Systems. ISPRS Journal of Photogrammetry and Remote Sensing 58 (3-4), 129-151.

Barber, D., Mills, J. and Smith-Voysey, S., 2008. Geometric Validation of a Ground-based Mobile Laser Scanning System. ISPRS Journal of Photogrammetry and Remote Sensing 63 (1), 128-141.

Barnard, S.T. and Thompson, W.B., 1980. Disparity Analysis of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (4), 333-340.

Bartoli, A., Hartley, R.I. and Kahl, F., 2003. Motion From 3D Line Correspondences: Linear and Non-Linear Solutions IEEE International Conference on Computer Vision and Patten Recognition (CVPR'03), madison, Wisconsin, USA, pp. 477-484.

Bay, H., Ess, A., Tuytelaars, T. and Gool, L.V., 2008. SURF: Speeded Up Robust Features. Computer Vision and Image Understanding 110 (3), 346-359.

Becker, S., Haala, N. and Fritsch, D., 2008. Combined Knowledge Propagation for Facade Reconstruction. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science XXXVII (Part B5), 423-430.

Birchfield, 2011. Kanade-Lucas-Tomasi (KLT) feature trackter. http://www.ces.clemson.edu/~stb/klt (accessed 2011.3.11).

Boldt, M., Weiss, E. and Riseman, E., 1989. Token-Based Extraction of Straight Lines. IEEE Transaction on Sustems,Man,and Cybernetics 19 (6), 1581-1594.

Bouguet, J.-Y., 2000. Pyramidal Implementation of the Lucus Kanade Feature Tracker. http://robots.stanford.edu/cs223b04/algo_tracking.pdf (accessed 2011.3.11).

Brenner, C., 2000. Towards Fully Automatic Generation of City Models. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science XXXIII (part B3), 85-92.

Brenner, C., 2005. Building Reconstruction from Images and Laser Scanning. International Journal of Applied Earth Observation and Geoinformation 6, 187-198.

Brenner, C., Dold, C. and Julge, K., 2003. Fusion, Interpretation and Combination of Geodata for the Extraction of Topographic Objects. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (3/W13).

Brown, D.C., 1966. Decentering distortion of lenses. Photogrammetric Engineering 7, 444-462.

Brunn, A. and Weidner, U., 1997. Extracting Buildings from Digital Surface Models. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 32 (3-4 W2).

Burns, J.B., Hanson, A.R. and Riseman, E.M., 1986. Extracting Straight Lines. IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6), 425-455.

Burt, P.J., 1988. Smart Sensing within a Pyramid Vision Machine. Proceedings of the IEEE 76 (8), 1006-1013.

Canny, J., 1986. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6), 679-698.

Clarke, J.C., Carlsson, S. and Zisserman, A., 1996. Detecting and Tracking Linear Features Efficiently. , British Machine Vision Conference, http://www.bmva.org/bmvc/1996/ (accessed 2011.3.11).

Colombari, A., Fusiello, A. and Murino, V., 2007. Video Objects Segmentation by Robust Background Modeling, the 14th International Conference on Image Analysis and Processing Washington, DC, USA, pp. 155-164.

Cornelis, N., Leibe, B., Cornelis, K. and Van Gool, L., 2008. 3D Urban Scene Modeling Integrating Recognition and Reconstruction. International Journal of Computer Vision 78 (2), 121-141.

Cristani, M., Bicego, M. and Murino, V., 2002. Integrated Region- and Pixelbased Approach to Background Modelling, IEEE Workshop on Motion and Video Computing, pp. 3-8.

Cross, A.D.J., Wilson, R.C. and Hancock, E.R., 1997. Inexact Graph Matching Using Genetic Search. Pattern Recognition 30 (6), 953-970.

D'Apuzzo, N., 2003. Surface Measurement and Tracking of Human Body Parts from Multi Station Video Sequences. PhD thesis, ETH Zurich, Zurich, 147 pp.

Debevec, P., Taylor, C. and Malik, J., 1996. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach, SIGGRAPH'96, pp. 11-20.

Dick, A., Torr, P. and Cipolla, R., 2004. Modelling and Interpretation of Architecture from Several Images International Journal of Computer Vision 60 (2), 111-134.

Dobbert, T., 2005. Matchmoving: The Invisible Art of Camera Tracking. Sybex Inc., USA.

Du, Y., Cihlar, J., Beaubien, J. and Latifovic, R., 2001. Radiometric Normalization, Composition, and Quality Control for Satellite High Resolution Image Mosaics over Large Area. IEEE Transactions on Geoscience and Remote Sensing 39 (3), 623-634.

Egenhofer, M.J. and Franzosa, R.D., 1991. Point-set Topological Spatial Relations International Journal of Geographical Information Systems 5 (2), 161-174.

Egenhofer, M.J. and Herring, J.R., 1990. A Mathematical Framework for the Definition of Topological Relationships, the Fourth International Symposium on Spatial Data Handling, Zurich, Switzland, pp. 803-813.

El-Hakim, S., Whiting, E. and Gonzo, L., 2005a. 3D Modeling with Reusable and Integrated Building Blocks, The 7th Conference on Optical 3-D Measurement Techniques, pp. 3-5.

El-Hakim, S., Whiting, E., L. Gonzo and Girardi, S., 2005b. 3-D Reconstrction of Complex Architectures from Multiple Data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI (5/W17).

Ellul, C. and Haklay, M., 2006. Requirements for Topology in 3D GIS. Transaction in GIS 10 (2), 157-175.

Englert, R. and Gülch, E., 1996. One-Eye Stereo System for the Acquisition of Complex 3D Building Descriptions. GIS Journal 4 (6).

Eos Systems, 2011. Homepage PhotoModeler software. http://www.photomodeler.com (accessed 2011.3.11).

Förstner, W., 1986. A Feature based Correspondence Algorithm for Image Matching. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 26 (3), 150-166.

Förstner, W., 1998. On the Theoretical Accuracy of Multi Image matching, Restorration and Triangulation, Festschrift zum 65. Geburtstag von Prof. Dr.-Ing. mult. G. Konecny, Institut fur Photogrammetrie, Universitat Hannover

Förstner, W., 1999. 3D-City Models: Automatic and Semiautomatic Acquisition Methods. In: D. Fritsch and R. Spiller (Editors), Photogrammetry Week'99, pp. 291-303.

Förstner, W., 2002. Computer Vision and Photogrammetry --- Mutual Questions: Geometry, Statistics and Cognition, International Symposium Photogrammtry and Remote Sensing, pp. 151-164.

Faugeras, O., Fua, P., Hotz, B., Ma, R., Robert, L., Thonnat, M. and Zhang, Z., 1992a. Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms, International Workshop on Robust Computer Vision: Quality of Vision Algorithms, Karlsruhe, Germany, pp. 1-26.

Faugeras, O., Luong, Q.-T. and Maybank, S., 1992b. Camera Self-calibration: Theory and Experiments, Second European Conference on Computer Vision (ECCV'92), Santa Margherita Ligure, Italy, pp. 321-334.

Feng, W., 2002. Close-range photogrammetry. Wuhan University Press, Wuhan, China (in chinese).

Fisher, M.A. and Bolles, R.C., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography. Communications of ACM 24 (6), 381-395.

Fitzgibbon, A. and Zisserman, A., 1998. Automatic 3D Model Acquisition and Generation of New Images from Video Sequences, the European Signal Processing Conference, Rhodes, Greece, pp. 1261-1269.

Früh, C., Jain, S. and Zakhor, A., 2005. Data Processing Algorithms for Generating Textured 3D Building Façade Meshes From Laser Scans and Camera Images. International Journal of Computer Vision 61 (2), 159-184.

Gülch, E., 1991. Results of Test on Image Matching. ISPRS Journal of Photogrammetry and Remote Sensing 46 (1), 1-18.

Galo, M. and Tozzi, C.L., 2004. Feature-point based Matching: a Sequential Approach based on Relaxation Labeling and Relative Orientation. Journal of WSCG 12 (1), 113-120.

Google, 2011. Homepage Sketchup software. http://sketchup.google.com/ (accessed 2011.3.11).

Grün, A., 1985. Adaptive Least Squares Correlation: A Powerful Image Matching Technique. South Africa Journal of Photogrammetry, Remote Sensing and Cartography 14 (3), 175-187.

Grün, A. and Baltsavias, E.P., 1988. Geometrically Constrained Multiphoto Matching. Photogrammetric Engineering & Remote Sensing 54 (5), 633-641.

Grimson, W.E.L., 1985. Computational Experiments with a Feature Based Stereo Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 7 (1), 17-34.

Gutchess, D., Trajkovic, M., Cohen-Solal, E., Lyons, D. and Jain, A., 2001. A Background Model Initialization Algorithm for Video Surveillance, IEEE International Conference on Computer Vision, pp. 733-740.

Harris, C. and Stephens, M., 1988. A Combined Corner and Edge Detector, 4th Alvey Vision Conference, Manchester, UK, pp. 147-151.

Hartley, R., 1992. Estimation of Relative Camera Positions for Uncalibrated Cameras, Second European Conference on Computer Vision (ECCV'92), Santa Margherita Ligure, Italy, pp. 579-587.

Hartley, R.I. and Zisserman, A., 2004. Multiple View Geometry in Computer Vision (second edition). Cambridge University Press.

Harville, M., 2002. A Framework for High-level Feedback to Adaptive, Perpixel, Mixture-of-gaussian Background Models, European Conference on Computer Vision, pp. 543-560.

Hays, J. and Efros, A.A., 2007. Scene Completion Using Millions of Photographs. ACM Transaction on Graphics (SIGGRAPH 2007) 26 (3).

Heinrichs, M., Hellwich, O. and Rodehorst, V., 2008. Robust Spatio-temporal Feature Tracking. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences VXXXVII (Part B3a), 51-56.

Heipke, C., 1996. Overview of Image Matching Techniques, OEEPE Workshop on the Application of Digital Photogrammetric Workstations, Lausanne, Switzerland, pp. 173-191.

Helava, U.V., 1988. Object-Space Least-Squares Correlation. Photogrammetric Engineering & Remote Sensing 54 (6), 763-769.

Henricsson, O. and Baltsavias, E., 1997. 3-D Building Reconstruction with ARUBA: A Qualitative and Quantitative Evaluation. In: A. Gruen, E. Baltsavias and O. Henricsson (Editors), Automatic Extraction of Man-Made Objects from Aerial and Space Images (II), Birkhauser Verlag, pp. 65-76.

Herley, C., 2005. Automatic Occlusion Removal from Minimum Number of Images. IEEE International Conference on Image Processing 2, 1046-1049.

Heuel, S., 2004. Uncertain Projective Geometry: statistical reasoning from pholyhedral object reconstruction. PhD, University of Bonn, Germany.

Heuel, S. and Förstner, W., 2001. Topological and Geometrical Models for Building Extraction from Multiple Images. In: E.P. Baltsavias, A. Grün and L. VanGool (Editors), Automatic Extraction of Man-Made Objects from Aerial and Satellite Images III. A.A.Balkema Publishers, Ascona, Switzerland, pp. 13-24.

Hilton, A., 2005. Scene Modelling from Sparse 3D Data. Image and Vision Computing 23 (10), 900-920.

Hough, P.V.C., 1962. Method and Means for Recognizing Complex Patterns. United States Patent 3069654

Hung, Y.P., Chen, C.S., Hung, K.C., Chen, Y.S. and Fuh, C.S., 1998. Multipass Hierarchical Stereo Matching for Generation of Digital Terrain Models from Aerial Images. Machine Vision and Applications 10 (5-6), 280-291.

Illingworth, J. and Kittler, J., 1987. The Adaptive Hough Transform. IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (5), 690-698.

Irani, M., 2002. Multi-Frame Correspondence Estimation Using Subspace Constraints. International Journal of Computer Vision 48 (3), 173-194.

Jain, S., 2002. Hole Filling in Images. MSc thesis, University of California at Berkeley.

Javed, O., Shafique, K. and Shah, M., 2002. A Hierarchical Approach to Robust Background Subtraction Using Color and Gradient Information, IEEE Workshop on Motion and Video Computing (MOTION' 02).

Jazayeri, I. and Fraser, C.S., 2010. Interest Operators for Feature-based Matching in Close Range Photogrammetry. The Photogrammetric Record 25 (129), 24-41.

Jiang, W., Zheng, S., Zhang, Z. and Zhang, J., 2003. Feature-based Image Matching for Aerial Images. Geomatics and Information Science of Wuhan University 28 (5), 510-513.

Kang, Z., 2007. The Automatic Removal of Tree Occlusion on Building Facade. Jounal of Shandong University of Science and Technology 26 (1), 27-29.

Kim, K., Chalidabhongse, T.H., Harwood, D. and Davis, L., 2005. Real-time Foreground-background Segmentation Using Codebook Model. Real-Time Imaging 11 (3), 172-185.

Klein, G. and Murray, D., 2006. Full-3D Edge Tracking with a Particle Filter, British Machine Vision Conference (BMVC'06), Edinburgh, pp. 1119--1128.

Koch, R., Pollefeys, M. and Van Gool, L., 2000. Realistic Surface Reconstruction of 3D Scenes from Uncalibrated Image Sequences. Journal of Visualization and Computer Animation 11 (3), 115-127.

Kunii, Y. and Chikatsu, H., 2004. Efficient Line Matching by Image Sequential Analysis for Urban Area Modelling. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 35 (B8), 211-214.

Lee, D.-S., Hull, J.J. and Erol, B., 2003. A Bayesian Framework for Gaussian Mixture Background Modeling, IEEE International Conference on Image Processing, pp. 973-976.

Levin, A., Zomet, A., Peleg, S. and Weiss, Y., 2004. Seamless Image Stitching in the Gradient Domain, The eighth European Conference on Computer Vision (ECCV2004), pp. 377-389.

Li, H., Lavin, M.A. and LeMaster, R.J., 1986. Fast Hough Transform: a Hierarchical Approach. Computer Vision Graphics and Image Processing 36 (2-3), 139-161.

Longuet-Higgins, H.C., 1981. A Computer Algorithm for Reconstructing a Scene from Two Projections. Nature 293, 133-135.

Lowe, D.G., 1999. Object Recognition from Local Scale-Invariant Features, International Conference on Computer Vision (ICCV), Corfu, Greece, pp. 1150-1157.

Lowe, D.G., 2004. Distinctive Image Features from Scale-invariant Keypoints. International Journal of Computer Vision 60 (2), 91-110.

Lucas, B.D. and Kanade, T., 1981. An Iterative Registration Technique with an Application to Stereo Vision, International Joint Conference on Arificial Intelligence pp. 674-679.

Maas, H.G. and Kersten, T., 1997. Aerotriangulation and DEM/Orthophoto Generation from High Resolution Stillvideo Imagery. Photogrammetric Engineering & Remote Sensing 63 (9), 1079-1084.

Marapane, S.B. and Trivedi, M.M., 1994. Multi-primitive hierarchical (MPH) stereo analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (6), 227-240.

Marr, D. and Poggio, T., 1979. A Computational Theory of Human Stereo Vision, Proceedings of the Royal Society of London,Series B, Biological Sciences, pp. 301-328.

Mayer, H., 1999. Automatic Object Extraction from Aerial Imagery - A Survey Focusing on Buildings. Computer Vision and Image Understanding 74 (2), 138-149.

Mayer, H., 2008. Object Extraction in Photogrammetric Computer Vision. ISPRS Journal of Photogrammetry and Remote Sensing 63 (2), 613-622.

Mayer, H. and Reznik, S., 2006. MCMC Linked With Implicit Shape Models and Plane Sweeping for 3D Building Facade Interpretation in Image Sequences, Photogrammetric Computer Vision (PCV'06).

Mayer, H. and Reznik, S., 2007. Building Facade Interpretation from Uncalibrated Wide-baseline Image Sequences. ISPRS Journal of Photogrammetry and Remote Sensing 61 (6), 371-380.

Mayhew, J.E.W. and Frisby, J.P., 1981. Psychophysical and Computational Studies Towards a Theory of Human Stereopsis. Artificial Intelligence 17, 349-385.

McGlone, J.C., Mikhail, E.M., Bethel, J. and Mullen, R., 2004. Manual of Photogrammetry. American Society for Photogrammetry and Remote Sensing.

Meer, P. and Georgescu, B., 2001. Edge Detection with Embedded Confidence. Pattern Analysis and Machine Intelligence, IEEE Transactions on 23 (12), 1351-1365.

Mikolajczyk, K. and Schmid, C., 2004. Scale and Affine Invariant Interest Point Detectors. International Journal of Computer Vision 60 (1), 63-86.

Moravec, H.P., 1977. Towards Automatic Visual Obstacle Avoidance, 5th International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, USA, pp. 584.

Mortenson, M.E., 1997. Geometric Modeling. John Wiley & Sons, New York.

Mukunoki, M., Yasuda, K. and Asada, N., 2005. 3D Model Generation from Image Sequences Using Global Geometric Constraint, Advances in Visual Computing, pp. 470-477.

Nevatia, R. and Babu, K., 1979. Linear Feature Extraction and Description, International Joint Conference on Artificial Intelligence Tokyo, Japan, pp. 639-641.

Nistér, D., 2004a. Automatic Passive Recovery of 3D from Images and Video, 2nd IEEE International Symposium on 3D Data Processing,Visualization and Transmission, Thessaloniki, Greece, pp. 438–445.

Nistér, D., 2004b. An Efficient Solution to the Five-point Relative Pose Problem. IEEE Transactions on Pattern Analysis Machine Intelligence 26 (6), 756-770.

Okutomi, M. and Kanade, T., 1993. A Multiple-baseline Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (4), 353-363.

Open Geospatial Consortium, 2008. OpenGIS City Geography Markup Language (CityGML) Encoding standard. OGC 08-007r1, Version 1.0.0.

Pigot, S., 1991. Topological models for 3D spatial information systems, 10th International Symposium on Computer Assisted Cartography, Baltimore, Maryland, USA, pp. 369-391.

Pilouk, M., 1996. Integrated Modelling for 3D GIS. PhD thesis, International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede.

Pollefeys, M., Koch, R., Vergauwen, M. and Van Gool, L., 2000. Automated Reconstruction of 3D Scenes from Sequences of Images. ISPRS Journal of Photogrammetry and Remote Sensing 55 (4), 251-267.

Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G. and Towles, H., 2008. Detailed Real-Time Urban 3D Reconstruction from Video. International Journal of Computer Vision 78 (2), 143-167.

Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J. and Koch, R., 2004. Visual Modelling with a Hand-held Camera. International Journal of Computer Vision 59 (3), 207-232.

Porikli, F. and Tuzel, O., 2003. Human Body Tracking by Adaptive Background Models and Mean-shift Analysis, IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS), http://www.merl.com/publications/TR2003-036/ (accessed 2011.03.11).

Pu, S. and Vosselman, G., 2009. Knowledge based Reconstruction of Building Models from Terrestrial Laser Scanning Data. ISPRS Journal of Photogrammetry and Remote Sensing 64 (6), 575-584.

Remondino, F., 2006. Detectors and Descriptors for Photogrammetric Applications, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 49-54.

Remondino, F. and El-Hakim, S., 2006. Image-Based 3D Modeling: A Review. The Photogrammetric Record 21 (115), 269-291.

Roman, A., Garg, G. and Levoy, M., 2004. Interactive Design of Multi-perspective Images for Visualizing Urban Landscapes, Proceedings of the conference on Visualization '04 pp. 537-544.

Rosenfeld, A., Hummel, R. and Zucker, S., 1976. Scene Labeling Using Relaxation Operation. IEEE Transactions on Systems, Man and Cybernetics 6 (6), 420-433.

Rosten, E. and Drummond, T., 2006. Matching Learning for High-speed Corner Detection, Proceedings of the European Conference on Computer Vision, Graz, Austria, pp. 430-443.

Sato, T., Kanbara, M. and Yokoya, N., 2003. Outdoor Scene Reconstruction from Multiple Image Sequences Captured by a Hand-held Video Camera, IEEE Conference on Multisensor Fusion and lntegration for Intelligent Systems pp. 113-118.

Schenk, T., 1999. Digital Photogrammetry TerraScience, Laurelville, 428 pp.

Schindler, G., Dellaert, F. and Sing Bing, K., 2007. Inferring Temporal Order of Images From 3D Structure, IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07) pp. 1-7.

Schmid, C., Mohr, R. and Bauckhage, C., 2000. Evaluation of Interest Point Detectors. International Journal of Computer Vision 37 (2), 151-172.

Schmid, C. and Zisserman, A., 1997. Automatic Line Matching across Views, the Conference on Computer Vision and Pattern Recognition (CVPR'97), Washington DC, USA, pp. 666-671.

Scholze, S., Moons, T. and Gool, L.V., 2002. A Probabilistic Approach to Building Roof Reconstruction Using Semantic Labeling, Lecture Notes In Computer Science. Springer, London, UK, pp. 257-264.

Shi, J. and Tomasi, C., 1994. Good Features to Track, IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600.

Shum, H.Y. and Kang, S.B., 2000. A Review of Image-based Rendering Techniques, IEEE/SPIE Visual Communications and Image Processing (VCIP), Perth, Australia, pp. 2-13.

Sinha, S.N., Frahm, J.-M., Pollefeys, M. and Genc, Y., 2007. Feature Tracking and Matching in Video Using Programmable Graphics Hardware. Machine Vision and Applications doi:10.1007/s00138-007-0105-z.

Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M. and Pollefeys, M., 2008. Interactive 3D Architectural Modeling from Unordered Photo Collections. ACM Transactions on Graphics 27 (5), Article No. 159.

Smith, S. and Brady, J., 1997. SUSAN: A New Approach to Low-level Image Processing. International Journal of Computer Vision 23 (1), 45-78.

Snavely, N., Seitz, S. and Szeliski, R., 2008. Modeling the World from Internet Photo Collections. International Journal of Computer Vision 80 (2), 189-210.

Soodamani, R. and Liu, Z.Q., 1998. A Novel Fuzzy Hough Transform for Shape Representation, IEEE International Conference on Fuzzy Systems, pp. 1605-1608.

Suveg, D. and Vosselman, G., 2004. Reconstruction of 3D Building Models from Aerial Images and Maps. ISPRS Journal of Photogrammetry and Remote Sensing 58 (3-4), 202-224.

Szeliski, R. and Kang, S.B., 1994. Recovering 3D Shape and Motion from Image Streams Using Non-Linear Least Squares. Journal of Visual Communication and Image Representation 5 (1), 10-28.

Taillandier, F. and Deriche, R., 2004. Automatic Buildings Reconstruction from Aerial Images: a Generic Bayesian Framework. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 35 (B3).

Tang, L., Tsui, H.T. and Wu, C.K., 2002. Dense stereo matching based on propagation with a voronoi diagram, India Conference on Computer Vision, Graphics and Image Processing III, Ahmedabad, India, pp. 230-240.

Tao, V.C., 2000. Semi-Automated Object Measurement Using Multi-Image Matching from Mobile Mapping Image Sequences. Photogrammetric Engineering & Remote Sensing 67 (12), 1347-1357.

Tarsha-Kurdi, F., Landes, T. and Grussenmeyer, P., 2007. Hough-transform and Extended Ransac Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3/W52), 407-412.

Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M. and Master, N., 2003. Calibrated, Registered Images of an Extended Urban Area. International Journal of Computer Vision 53 (1), 93-107.

Theobalt, C., Magnor, M., Schuler, P. and Seidel, H.-P., 2002. Combining 2D Feature Tracking and Volume Reconstruction for Online Video-based Human Motion Capture, 10th Pacific Conference on Computer Graphics and Applications, pp. 96-103.

Tissainayagam, P. and Suter, D., 2004. Assessing the Performance of Corner Detectors for Point Feature Tracking Applications. Image and Vision Computing 22 (8), 663-679.

Tomasi, C. and Kanade, T., 1992. Shape and Motion from Image Streams:a Factorization Method. International Journal of Computer Vision 9 (2), 137-154.

Torresani, L. and Hertzmann, A., 2004. Automatic Non-Rigid 3D Modeling from Video, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 299-312.

Toyama, K., Krumm, J., Brumitt, B. and Meyers, B., 1999. Wallflower: Principles and Practice of Background Maintenance, International Conference on Computer Vision, pp. 255-261.

Tsai, F., Chen, C.-H., Liu, J.-K. and Hsiao, K.-H., 2006. Texture Generation and Mapping Using Video Sequences for 3D Building Models, Lecture Notes in Geoinformation and Cartography. Springer, Berlin Heidelberg, pp. 429-438.

Tsai, R.Y. and Huang, T.S., 1984. Uniqueness and Estimation of Three Dimensional Motion Parameters of Rigid Objects with Curved Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1), 13-27.

Ullman, S., 1979. The Interpretation of Visual Motion. MIT Press, Cambridge, Massachusetts.

Uyttendaele, M., Eden, A. and Szeliski, R., 2001. Eliminating Ghosting and Exposure Artifacts in Image Mosaics. IEEE Conference on Computer Vision and Pattern Recognition 2, 509-516.

Van den Heuvel, F.A., 1999. Line-photogrammetric Mathematical Model for the Reconstruction of Polyhedral Objects, Proceedings of SPIE Videometrics VI, pp. 60-71.

Vicon, 2011. Homepage Boujou software. http://www.vicon.com/boujou/ (accessed 2011.3.11).

Vosselman, G., 1998. Interactive Alignment of Parametrised Object Models to Images. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 32 (3), 272-278.

Vosselman, G. and Dijkman, S., 2001. 3D Building Model Reconstruction from Point Clouds and Ground Plans. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (3/W5), 37-44.

Vosselman, G., Gorte, B.G.H., Sithole, G. and Rabbani, T., 2004. Recognising structure in laser scanner point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (8/W2), 33-38.

Warrick, A.L. and Delaney, P.A., 1997. Detection of Linear Features Using a Localized Radon Transform, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2769-2772.

Wen, G. and Wang, R., 2001. A Robust Approach to Extracting Straight Lines. Journal of Software 12 (11), 1660-1666(in chinese).

Werner, T. and Zisserman, A., 2002. New Techniques for Automated Architectural Reconstruction from Photographs, Seventh European Conference on Computer Vision, pp. 541-555.

Wilczkowiak, M., Brostow, G.J., Tordoff, B. and Cipolla, R., 2005a. Hole Filling though Photomontage, The 16th British Machine Vision Conference (BMVC'05), Oxford, UK.

Wilczkowiak, M., Sturm, P. and Boyer, E., 2005b. Using Geometric Constraints though Parallelepipeds for Calibration and 3D Modeling IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2), 194-207.

Wixson, L., 2000. Detecting Salient Motion by Accumulating Directionally-consistent Flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 744-780.

Xiao, J., Fang, T., Zhao, P., Maxime, L. and Quan, L., 2009. Image-based street-side city modeling. ACM Transaction on Graphics 28 (5), 1-12.

Xu, S., 2007. Straight Line Extraction Method for Stereoimage Feature Matching. PhD thesis, Wuhan University, Wuhan, 122 pp.

Zhang, L., Shen, W., Zhang, Z. and Jianqing, Z., 2000. Neural Network Image Matching Based on Variable Weight Smooth Constraint. Journal of Wuhan Technical University of Surveying and Mapping 25 (1), 55-59.

Zhang, Y., Zhang, Z., Zhang, J. and Wu, J., 2005. 3D Building Modelling with Digital Map, Lidar Data and Video Image Sequences. The Photogrammetric Record 20 (111), 285-302.

Zhang, Z., Wu, J. and Zhang, J., 2004. A Method for Straight Line Extraction Based on Line Space. Geomatics and Information Science of Wuhan University 29 (3).

Zhang, Z., Yang, S., Zhang, J. and Ke, T., 2007. Multi-Baseline Digital Close-Range Photogrammetry. Geospatial Information 5 (1), 1-4.

Zhong, J. and Sclaroff, S., 2003. Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter. IEEE International Conference on Computer Vision (ICCV 2003) 2.

Zhu, Q., Zhao, J., Lin, H. and Gong, J.Y., 2005. Triangulation of Well-defined Points as a Constraint for Reliable Image Matching. Photogrammetric Engineering and Remote Sensing 71 (9), 1063-1069.

Ziou, D. and Tabbone, S., 1998. Edge Detection Techniques - An Overview. International Journal of Pattern Recognition and Image Analysis 8 (4), 537-559.

Zisserman, A., Werner, T. and Schaffalitzky, F., 2001. Towards Automated Reconstruction of Architectural Scenes from Multiple Images, 25th Workshop of the Austrian Association for Pattern Recognition, Berchtesgaden, Germany, pp. 9-23.

List of Figures

List of Tables

List of Publication

Yixiang Tian, Markus Gerke, George Vosselman, Qing Zhu, 2010, Knowledge-based building reconstruction from terrestrial video sequences, ISPRS Journal of Photogrammetry and Remote Sensing, 65(4), 395-408

Yixiang Tian, Markus Gerke, George Vosselman, Qing Zhu, 2009, Knowledge-based topological reconstruction for building facade surface patches, the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, 38(5/W1) (on CD_ROM)

Yixiang Tian, Markus Gerke, George Vosselman, Qing Zhu, 2009, Automatic Surface Patch Generation from a Video Image Sequence, 3D Geo-Information Sciences: requirements, acquisition, modeling, analysis, visualization, Springer, Berlin, 235-246

Lihua Zhang, Qing Zhu, Li Zhang, Deqing Liang and Yixiang Tian, 2009, An Efficient Algorithm to Plot Flooded Intertidal Areas, Computers & Geosciences, 35(6), 1072-1078

Yixiang Tian, Markus Gerke, George Vosselman, Qing Zhu, 2008, Automatic Edge Matching Across an Image Sequence Based on Reliable Points, the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, 37(B3b), 657-662

Qing Zhu, Bo Wu, Yixiang Tian, 2007, Propagation strategies for stereo image matching based on the dynamic triangle constraint, ISPRS Journal of Photogrammetry and Remote Sensing, 62(4), 295-308

Curriculum Vitae

Yixiang Tian was born in Wuhan, China in 1981. From 1999 to 2003, she studied geographic information system at the China University of Geosciences (Wuhan). After receiving a Bachelor's degree in 2003, she enrolled for a Master's degree at the Wuhan University, with a specialization in photogrammetry and remote sensing. Her research was on topographical feature extraction and image matching. As an outstanding MSc student she was authorized to continue the study for PhD degree without defending the MSc thesis. November 2006 she started a PhD study at the International Institute for Geo-Information Science and Earth Observation (ITC), on the topic of building reconstruction from terrestrial video image sequences. Since February 2011, she holds a lecture position at the department of Surveying and Geo-informatics at Tongji University, China. Her main research and education responsibility are (semi-) automated 3D information reconstruction and sustainable development applications.