3D Reconstruction from Video Using a Mobile Robot.

A. Manessis

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
School of Electronic Engineering, Information Technology and Mathematics
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

December 2001

© A. Manessis 2001
Summary

An autonomous robot able to navigate inside an unknown environment and reconstruct full 3D scene models using monocular video has been a long term goal in the field of Machine Vision. A key component of such a system is the reconstruction of surface models from estimated scene structure. Sparse 3D measurements of real scenes are readily estimated from N-view image sequences using structure-from-motion techniques. In this thesis we present a geometric theory for reconstruction of surface models from sparse 3D data captured from N camera views. Based on this theory we introduce a general N-view algorithm for reconstruction of 3D models of arbitrary scenes from sparse data. Using a hypothesise and verify strategy this algorithm reconstructs a surface model which interpolates the sparse data and is guaranteed to be consistent with the feature visibility in the N-views. To achieve efficient reconstruction independent of the number of views a simplified incremental algorithm is developed which integrates the feature visibility independently for each view. This approach is shown to converge to an approximation of the real scene structure and have a computational cost which is linear in the number of views. Surface hypothesis are generated based on a new incremental planar constrained Delaunay triangulation algorithm. We present a statistical geometric framework to explicitly consider noise inherent in estimates of 3D scene structure from any real vision system. This approach ensures that the reconstruction is reliable in the presence of noise and missing data. Results are presented for reconstruction of both real and synthetic scenes together with an evaluation of the reconstruction performance in the presence of noise.

Key words: Scene modelling, sparse data, incremental reconstruction, Delaunay triangulation, geometric uncertainty, mobile robot.

Email: A.Manessis@eim.surrey.ac.uk
WWW: http://www.eim.surrey.ac.uk/
Acknowledgements

First I would like to thank both my supervisors Adrian Hilton and Phil Palmer for giving me the opportunity to embark on this exciting research area. I especially want to express my appreciation to Adrian for all the useful suggestions he had to offer whenever I thought that nothing further could be done for solving a problem. The significance of good supervision towards the successful completion of a PhD is known. However, equally important is the personal welfare and I feel lucky that I make so many friends over the years. To all of them I owe my gratitude as I know that to put up with me is a tough job!!!! At last I want to thank my parents and my sister for their support in every aspect. This study is dedicated to them.
Contents

Contents ................................................................. vii
List of Figures ........................................................... xii
List of Tables ............................................................. xiii

1 Introduction .............................................................. 1
   1.1 Environment modelling using an autonomous robot platform ...... 2
      1.1.1 Hardware configuration ............................................ 3
      1.1.2 Visual Reconstruction Software system ......................... 3
   1.2 Motivation and aim .................................................... 5
   1.3 Outline of the thesis .................................................. 7

2 Overview of related work on Scene Modelling .......................... 9
   2.1 Image-Based Modelling ................................................ 10
      2.1.1 Approximation of the Plenoptic Function ...................... 10
      2.1.2 View Interpolation ............................................... 12
   2.2 Geometry Modelling ................................................... 13
      2.2.1 Reconstruction using active sensors ............................ 14
      2.2.2 Reconstruction from dense passive sensors range data ........ 19
      2.2.3 Shape from Silhouettes ......................................... 21
      2.2.4 Reconstruction based on photo consistency ..................... 23
   2.3 Modelling from sparse data .......................................... 27
      2.3.1 Structure from motion ............................................. 27
      2.3.2 Single view reconstruction ....................................... 28
      2.3.3 Multiple view reconstruction ................................... 29
   2.4 Reconstructing representations for real world scenes ............... 32
3 **Theory of reconstruction from sparse data**  
3.1 Problem statement and definition of a consistent model  
3.2 Reconstruction of a Consistent 3D Scene Model  
3.3 Single-view Reconstruction  
3.4 N-view Reconstruction  
3.5 Proof of convergence  
3.6 N-View Reconstruction Algorithm Complexity  

4 **Incremental Reconstruction Algorithm**  
4.1 Reconstruction theory simplification  
4.2 Incremental Algorithm  
4.3 Algorithm implementation  
4.3.1 Feature Update  
4.3.2 Feature visibility  
4.3.3 Model Integration  
4.4 Recursive N-View Algorithm Complexity  

5 **Surface estimation using 2D constrained Delaunay triangulation**  
5.1 Delaunay Triangulations  
5.2 Constrained Delaunay Triangulation  
5.2.1 Point Insertion  
5.2.2 Line segment insertion  
5.2.3 Convergence of line segment insertion algorithm to CDT  
5.3 CDT complexity analysis  

6 **Robust Reconstruction**  
6.1 Geometric Uncertainty  
6.2 Model update based on uncertainty information  
6.2.1 Impose junctions  
6.2.2 Feature update and uncertainty propagation  
6.3 Visibility test using feature uncertainty  

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Problem statement and definition of a consistent model</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Reconstruction of a Consistent 3D Scene Model</td>
<td>39</td>
</tr>
<tr>
<td>3.3</td>
<td>Single-view Reconstruction</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>N-view Reconstruction</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Proof of convergence</td>
<td>44</td>
</tr>
<tr>
<td>3.6</td>
<td>N-View Reconstruction Algorithm Complexity</td>
<td>44</td>
</tr>
<tr>
<td>4.1</td>
<td>Reconstruction theory simplification</td>
<td>47</td>
</tr>
<tr>
<td>4.2</td>
<td>Incremental Algorithm</td>
<td>49</td>
</tr>
<tr>
<td>4.3</td>
<td>Algorithm implementation</td>
<td>52</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Feature Update</td>
<td>53</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Feature visibility</td>
<td>55</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Model Integration</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>Recursive N-View Algorithm Complexity</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>Delaunay Triangulations</td>
<td>61</td>
</tr>
<tr>
<td>5.2</td>
<td>Constrained Delaunay Triangulation</td>
<td>62</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Point Insertion</td>
<td>65</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Line segment insertion</td>
<td>66</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Convergence of line segment insertion algorithm to CDT</td>
<td>70</td>
</tr>
<tr>
<td>5.3</td>
<td>CDT complexity analysis</td>
<td>72</td>
</tr>
<tr>
<td>6.1</td>
<td>Geometric Uncertainty</td>
<td>75</td>
</tr>
<tr>
<td>6.2</td>
<td>Model update based on uncertainty information</td>
<td>76</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Impose junctions</td>
<td>79</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Feature update and uncertainty propagation</td>
<td>82</td>
</tr>
<tr>
<td>6.3</td>
<td>Visibility test using feature uncertainty</td>
<td>83</td>
</tr>
</tbody>
</table>
## 7 Application of Model Reconstruction Framework

7.1 Synthetic data ................................................................. 87
  7.1.1 Noise free synthetic sequences ...................................... 88
  7.1.2 Synthetic noisy scenes ................................................. 95
  7.1.3 Sequences with incomplete feature sets ............................ 110
7.2 Real Data ................................................................. 118

## 8 Discussion, Conclusions and Further Work

8.1 What is the next step? ...................................................... 124

## A Order independency of reconstruction theory

## B Example application of Constrained Delaunay Triangulation algorithm

  B.1 Point insertion example ................................................. 131
  B.2 Line segment insertion example ....................................... 134

## C Synthetic Sequences

## D Reconstruction of Curved Surfaces
# List of Figures

1.1 State of the art robots. (a) Honda humanoid ASIMO. (b) SONY entertainment robot AIBO ............................... 2
1.2 Our mobile robot platform “Cyclops” ................................. 4
1.3 Outline of the software configuration modules of the system ........... 5

2.1 Model reconstruction by recursively carving a solid volume ............... 19
2.2 Example failure of shape from silhouettes approach. For both (a),(b) the resulted reconstruction will be (c) ................. 22

3.1 Examples of consistent and inconsistent representations for a simple scene 38
3.2 Estimating surface topology among 3D features. Structure projected to image plane, triangulated and backprojected to 3D space. ............ 41
3.3 Example of order inconsistency of projecting 3D structure to different viewing directions. Line $L_2$ lies infront of $L_1,L_3$ and while the order from left to right in view 1 is $L_1,L_3,L_2$ for view 2 is $L_2,L_1,L_3$ ....... 42

4.1 Illustration of reconstruction for two consecutive views. (a),(b),(c) Step 2. (d),(e) Step 4. (f),(g) Step 5a. ................................. 50
4.2 Illustration of reconstruction for two consecutive views. (a),(b) Step 5b. (c),(d) Step 5c. (f),(g) Step 6. ................................. 51
4.3 (a) Scene from viewpoint $i$. Only T junctions are extracted. (b) Reconstructed model based on the available junction information. (c) New viewpoint reveals a Y-junction between already reconstructed structure. ................................. 55

5.1 The Voronoi and Delaunay structures of a set of points. Delaunay triangles are sketched with bold lines. ................................. 63
5.2 Constrained triangulation of set of points B,D,E and line segments AC. (a)As the circle through A,B,C bounds D and conversely the circle through A,D,C encompass B no triangulation of the input set can be achieved that satisfy the empty Delaunay circle criterion for the feature set. (b)Conforming Constrained Delaunay Triangulation by introducing new points F,G. (c)Constrained Delaunay Triangulation. ................................. 64
5.3 Example of imposing a single line segment as a constraint on an existing mesh ............................................. 69

6.1 Uncertainty representation for (a)Point $\mathbf{x}$ and (b)Line $l = (\mathbf{x}_1, \mathbf{x}_2)$ and point $\mathbf{p}$ on the line. .......................................................... 77

6.2 Uncertainty representation for triangle $t = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ ................................................................. 78

6.3 Example of noisy T junction between 3D lines $l_1, l_2$. (a)The image plane projections of the lines are intersected (b)Lines in 3D from a different angle. ........................................ 81

6.4 Update an existing line segment estimate $(p_1, p_2)$ to the new corresponding measurement $(x_1, x_2)$ ................................................................. 82

6.5 One dimensional visibility test. (a)Model’s triangles and features are tested for 2D overlap in the image plane. (b)Representation of 1D variance of $\mathbf{p}, \mathbf{p}_t$ along the projection optical ray. ........................................ 84

7.1 Top view maps for the synthetic scenes. Arrows represent the approximate positions of the cameras. ................................................................. 90

7.2 Reconstructed models for the room with the boxes sequence. (a),(b),(c),(d) Models up to 1,5,9 and 13 images processed. (e) Final reconstructed model from three different angles ........................................ 92

7.3 Reconstructed models for the two rooms sequence. (a),(b),(c),(d), (e) (f) Models up to 1,5,9 and 13 images processed. (e) Final reconstructed model from different viewing angles ........................................ 93

7.4 Reconstructed models for the floor sequence. (a),(b),(c),(d), (e) (f) Models up to 1,5,9 and 13 images processed. (e) Final reconstructed model from different viewing angles ........................................ 94

7.5 Processing time relative to the number of the frame in the sequence. Times has been measured on an 400MHz Ultra Enterprise Sun system and are rounded to the closest second ........................................ 95

7.6 Reconstruction of boxes sequence with noise level $r_1$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles. ........................................ 97

7.7 Reconstruction of boxes sequence with noise level $r_2$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles. ........................................ 98

7.8 Reconstruction of boxes sequence with noise level $r_3$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles. ........................................ 99
7.9 Reconstruction of boxes sequence with noise level $r_4$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles. .......................................................... 100

7.10 Reconstruction of the two rooms scene with noise level $r_1$. (a),(b),(c) Frames 0,6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1,6,15 frame respectively. (g),(h) Final model from different viewing angles. .......................................................... 101

7.11 Reconstruction of the two rooms scene with noise level $r_2$. (a),(b),(c) Frames 0,6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1,6,15 frame respectively. (g),(h) Final model from different viewing angles. .......................................................... 102

7.12 Reconstruction of the two rooms scene with noise level $r_3$. (a),(b),(c) Frames 0,6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1,6,15 frame respectively. (g),(h) Final model from different viewing angles. .......................................................... 103

7.13 Reconstruction of the two rooms scene with noise level $r_4$. (a),(b),(c) Frames 0,6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1,6,15 frame respectively. (g),(h) Final model from different viewing angles. .......................................................... 104

7.14 Models geometry evaluation for the boxes sequence. The mean, variance and maximum distances of estimated models relative to the ground truth information .......................................................... 107

7.15 Models geometry evaluation for the two rooms sequence. The mean, variance and maximum distances of estimated models relative to the ground truth information .......................................................... 108

7.16 Example of reconstruction failure. (a),(b),(c),(d) Reconstructed models up to frames 0,4,5 and 15 respectively. .......................................................... 109

7.17 Example of reconstruction failure. (a) Reconstruction up to frame 5. (b) Frame 6 with superimposed features. (c) Model after frame 6. (d) Final model .......................................................... 109

7.18 A wire-frame representation of the seq. 1 with line features enumerated 111

7.19 Reconstructed model with no junction information. (a),(b),(c) Final model from several viewing positions. (d) A close look at the reconstructed boxes.112

7.20 Reconstruction where significant line structure is missing from the input feature set. (a),(b),(c) Model viewed from different angles. ............................. 113

7.21 Sequence of frames for reconstruction when 20% of line features are hidden 115

7.22 Sequence of frames for reconstruction when 30% of line features are hidden 116

7.23 Reconstructed model from sequence with 20% of lines missing. (a),(b),(c) Models up to 2,9,16 frame respectively. (d),(e),(f) Final model. ......... 117
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.24</td>
<td>Reconstructed model from sequence with 30% of lines missing. (a),(b),(c) Models up to 0,8,18 frame respectively. (d),(e),(f) Final model.</td>
</tr>
<tr>
<td>7.25</td>
<td>Real sequence from the corner of a lab</td>
</tr>
<tr>
<td>7.26</td>
<td>Reconstruction of the real scene sequence. (a),(b),(c) Models up to 4,6 and 9 frame</td>
</tr>
<tr>
<td>B.1</td>
<td>Point insertion example. Lines in the stack have been sketched with dots</td>
</tr>
<tr>
<td>B.2</td>
<td>Line segment insertion example. Filled lines corresponds to swapped edges, dashed lines to the edges in the list waiting to be processed and dotted lines to edges in the stack</td>
</tr>
<tr>
<td>C.1</td>
<td>Frames from the room with boxes synthetic sequence</td>
</tr>
<tr>
<td>C.2</td>
<td>Frames from the 2 empty rooms sequence</td>
</tr>
<tr>
<td>C.3</td>
<td>Frames from the floor sequence (0-25)</td>
</tr>
<tr>
<td>C.4</td>
<td>Frames from the floor sequence (26-54)</td>
</tr>
<tr>
<td>D.1</td>
<td>Frames from the room with the cylinder sequence</td>
</tr>
<tr>
<td>D.2</td>
<td>Reconstruction of the cylinder sequence. (a),(b),(c),(d) Models up to 0,4,7 and 10 frame respectively. (e),(f),(g) Final model and close views of the cylinder</td>
</tr>
</tbody>
</table>
List of Tables

7.1 Table of model reconstruction statistics without noise .......................... 91
7.2 Table of model reconstruction statistics for sequences 1,2 and four different level of noise on 20 distinct experiments ........................................ 106
Chapter 1

Introduction

In 1921 a new play premiered in Prague called the Rossum’s Universal Robots. The author was Karel Capek and the original novel had only been written a year earlier. According to the story, Rossum had invented a formula to reproduce man like machines able to be used as cheap labour. The machines called robots have all human characteristics but lack the ability to innovate. Eventually, however robots revolted against mankind and killed humans. The script would not make much impression to an audience today when people are familiar with robots. For its time though it resulted in much controversy and extended social discussions. This is considered the birth of the term robot a word that originates from the Czech “robotica” that means servitude or hard labour.

Eighty years later how far are we from discovering the secret of Rossum’s formula and creating an intelligent machine? The answer depends on the way intelligence is defined. State-of-the-art robots today may look like humans (figure 1.1(a)) or even friendly pets (figure 1.1(b)) but their behaviour is restricted to a limited predefined set of interactions with their environment. Surprisingly, perhaps the most “intelligent” robots today are those participating in various maze contests where the robots are initially placed at the entrance of a maze and autonomously search for the exit.

This may not be considered for many as significant progress, especially compared with advances in the related field of computing. In the sixties, when research towards intelligent machines originated, researchers were full of optimism that humans were soon
going to be served by smart robots. The complexity of the problem had then been profoundly underestimated. Creating intelligent robotic machines is an interdisciplinary subject that involves different aspects of sensing, manipulation and most importantly thinking which will integrate the sensory information into action.

Maybe the most prominent human sense is vision. Vision provides us with all the information necessary to understand and interact with our surrounding environment. Hence it is not a coincidence that this sense was one of the first that man has tried to incorporate into machines. Navigating inside a scene and building a model representation of it is fundamental in visual perception. This thesis attempts to offer a small step forward in this direction.

\section{Environment modelling using an autonomous robot platform}

Today much progress has been achieved on dedicated robots operating, most of the time, in industrial plants and performing a specific task. These applications however require only a limited understanding of the environment. One of the most challenging
aspects of Machine Vision is to be able to navigate an autonomous vehicle around an unknown environment, avoiding collisions with obstacles and building up a map of that scene using information from one or more cameras.

The research presented in this thesis is part of a larger research project aimed at developing an autonomous mobile robot platform to achieve these tasks. This thesis presents the framework and algorithms developed for constructing 3D surface models from sparse 3D feature location estimates. Sparse estimates of 3D feature location are reconstructed from image sequences obtained from a single camera mounted on a mobile robot platform. Models are built progressively as the robot moves around the scene. Potential applications of this system include virtual walkthroughs and general telepresence for domains such as entertainment, computer games, authentication of hazardous environments and visualisation of real-estate.

1.1.1 Hardware configuration

The mobile robot platform used, called “Cyclops”, is shown in figure 1.2. The base has three wheels coupled together and controlled by an embedded microprocessor. The platform has a single colour video camera with motorised focus and zoom mounted on a pan-and-tilt head. A processor inside the pan-and-tilt head controls both the movement of the camera and the parameters of the lens. Composite video output of the camera is broadcasted by a wireless link to a frame grabber on a remote workstation. The main control unit of the platform is an onboard Intel x386 processor. A 2-way radio communication channel links the mobile platform to a remote computer. The onboard controller redirects incoming instructions to the various pieces of hardware on the robot and transmits odometry and camera motion information back.

1.1.2 Visual Reconstruction Software system

The system has been developed to be independent of the type of robot platform used as far as possible. In this direction the whole software has been designed in a modular way, an outline of which is presented in Figure 1.3. The Control system provides the low-level control instructions for the robot base, pan-and-tilt head and camera parameters.
Chapter 1. Introduction

Mobile robot control is directed by a planning module that provides a navigation path based on feedback from the partially reconstructed 3D scene model up to a particular time instance.

The Vision system takes as input the video stream from the frame grabber and extracts the “significant” features (points and lines) from each frame. Odometry information is used to obtain the approximate camera location. Epipolar geometry is utilised to accomplish automatic feature matching between consecutive frames in the image sequence [85]. Feature matches are fed into a structure-from-motion (SFM) module which computes 3D estimates of scene feature and camera location based on a recursive bundle adjustment algorithm [60],[62]. This process results in a sparse set of 3D scene feature location estimates (points and lines) together with estimates of their uncertainty covariance. The problem, which is the central topic of this thesis, is then to reconstruct a model of the scene surfaces from the sparse feature data.

The Environment Modelling system, presented in this thesis, uses the sparse measurements estimated from the SFM to build a local model of the scene at each time instant as seen from the specific camera viewpoint. This model is subsequently integrated with
1.2. Motivation and aim

There have been several groups and projects that tried to develop similar systems in the past. An early attempt was the DROID 3D system [91],[38] that used a single calibrated camera mounted on a moving vehicle. More recently the European project VANGUARD aimed at scene model reconstruction from a hand-held uncalibrated cam-

In a complete system this partial global model would be feedback to the robot control system for navigation planning and viewpoint selection. Realisation of a complete system for autonomous navigation is an ongoing task. Currently manual intervention is required for viewpoint selection. This thesis presents a theoretical analysis of the problem of reconstruction from sparse data captured from multiple views and subsequent algorithm development for the incremental reconstruction of surface models.

Figure 1.3: Outline of the software configuration modules of the system

the existing 3D global environment model constructed up to the previous frame. An iterative integration algorithm based on a geometric reconstruction framework is introduced to integrate the local and previous global model to obtain a new global model [57],[55],[56]. This integration is based on a hypothesise and verify strategy using feature visibility and is shown to converge to a representation of the real scene surfaces. Resulting models are exported in Virtual Reality Modelling Language (VRML) format so that viewing is possible through a remote Internet connection.
era. Both these systems have successfully achieved impressive results for estimating the geometry of features in the real world scenes. Unfortunately the problem of approximating the corresponding scene topologies by reconstructing the surfaces that span the space between the cloud of computed 3D features has been almost completely ignored.

In recent years extensive research has focused on solving the structure-from-motion problem and significant progress has been made both in theoretical and practical terms. In contrast, very limited work has been done on surface reconstruction that connects the sparse feature set to approximate the scene surface topology. This open-problem, and the need to find a solution in order to realise a complete scene reconstruction system, motivated the research presented in this thesis.

The only research that has attempted to address this problem of estimating scene surface topology based only on a sparse feature set estimated from image sequences has been undertaken by Faugeras et al. [32]. Like the model reconstruction approach presented in this thesis their approach is also based on feature visibility in multiple views. However, their approach is limited to batch reconstruction, the effects of noise on the feature positions were not considered and the final model was in a volumetric triangular form difficult to texture or render. The limitations of their approach are therefore prohibitive for any practical application of the algorithm.

A methodology aiming to circumvent all the above limitations was the principal objective of this study. This thesis primarily focuses on automatic surface reconstruction of indoor scenes and reflects the work that has been done on the Environment Modelling module of the system presented in the last section. It is assumed that at each time instant corresponding to acquisition of an image frame, input is provided in the form of a set of estimates of sparse 3D scene feature locations and camera poses with their associated covariance estimates. Sparseness of the feature set is a typical characteristic of geometry reconstruction from monocular video sequences of indoor scenes due to the absence of visual features in many regions which can be reliably matched between consecutive image frames. This sparseness is in contrast to the dense estimates of 3D surface shape obtained from active sensing. An important characteristic of the approach developed is the ability to progressively build a triangulated surface model as
new scene structure becomes visible so that the resulting model at any time instant can be used to plan the next best viewing location. This was considered to be an important criteria for the development of such a system. The methodology developed also aims to be computationally efficient to allow the processing of extended image sequences. A primary requirement is that the reconstructed model should converge to a consistent approximation of the true scene surface topology as the number of processed frames increases. Finally, the system should be robust under the effects of noise inherent in every real visual reconstruction system.

1.3 Outline of the thesis

This thesis is organised as follows. Chapter 2 presents a literature survey of research related to the reconstruction of 3D models and focuses on techniques applicable to large scale scenes. In the subsequent chapter 3 a general theory for the problem of reconstructing from sparse 3D data captured from N camera views is proposed. A general algorithm using a hypothesise and verify strategy based on feature visibility is proposed and shown to converge to the true scene surfaces as the number of images increases. Computationally efficient application of this theory over extended video sequences is achieved by an incremental algorithm that is presented in the next chapter 4. New surface hypothesis is based on a new incremental constrained Delaunay triangulation algorithm that is following in chapter 5. In chapter 6 the reliability of the algorithm in the presence of noise and missing data is considered. Covariance information associated with the feature and camera location estimates is utilised to build an uncertainty envelope around the reconstruction so that model verification based on feature visibility becomes tolerant to noise. Chapter 7 presents a comprehensive evaluation of the methodology on multiple synthetic scenes with simulated noise and a limited real data set. Finally, the thesis is concluded with a summary of the achievements of this study and suggestions for further work.
Chapter 1. Introduction
Chapter 2

Overview of related work on Scene Modelling

There is no doubt that over the last decade everybody has experienced the rapid invasion of computerised technology in different areas of our day to day life. Applications reach from cinema and TV to entertainment games, art and medicine. The digital world seems so exciting but at the same time its synthetic creation is still apparent.

Several research disciplines like photogrammetry, computer graphics and computer vision have endeavoured for the creation of realistic scene representations. Photogrammetrists were the first that studied problems like camera calibration, image registration and bundle adjustment. Building on this basis 3D computer vision considered the problem of automatically reconstructing geometric descriptions of real world (or even synthetic) objects using data captured from several different type of sensors.

Computer graphics on the other hand focused on the inverse problem of synthesising images from geometric models such as the ones resulted from the 3D vision field. Using additional information on the surface reflectance properties and the scene illumination conditions realistic images are rendered. Recent approaches however address the issue of creating such views not based on an underlying geometric model but instead on a number of images.

Work on generating 3D representations for real objects in general is a topic that can-
not be covered in the limited length of a single chapter. Thus the main focus has been on different techniques that have been proposed in the literature in the field of reconstruction of realistic large scale scenes. These techniques have been categorised as image-based and geometric-based. The primary criterion of this classification is the underlying structures used to build the reconstructed representation and it is either a set of images or geometric primitives.

2.1 Image-Based Modelling

Long before computers were invented artists and engineers, especially architects, were able to manually generate views of a scene having available not more than a few reference images. The automation of this intuitive ability is the target of today’s image based modelling techniques.

In the image based approach the world is modelled in terms of a number of source images which are used to synthesise novel images that represent the scene from arbitrary viewpoints. As images can capture even the fine details of appearance, the complexity of the modelling process is decoupled from the complexity of the scene. Thus the amount of photorealism in the generated images only depends on the quality of the input views.

Image based modelling has evolved from research in both computer vision and graphics fields. Although the majority of the work in this area has focused on direct view interpolation between example images another category of methods approximate the plenoptic function [1] and directly synthesise images from arbitrary viewpoints. These two main categories are further discussed in the following section.

2.1.1 Approximation of the Plenoptic Function

People visualise the world as an interaction of light with the surface of the objects surrounding them. A set of light rays passing through the eye and focusing on the retina results to an image of the environment. In this sense everything that can be seen is contained in the dense array of different light rays that fill the space.
Adelson et al. [1] first introduced the plenoptic function to describe the structure of light. Measuring this function involves deploying a camera at every possible position in the 3 dimensional space and recording the light intensity passing through the camera lenses at every possible angle for every possible wavelength at every time. Once this 7D function has been captured generating images of a scene from arbitrary positions reduces to a trivial ray indexing process.

The plenoptic function is in practice impossible to compute. However, under certain assumptions its dimensionality can be reduced. Two techniques that utilised a 4D subset of the plenoptic function were the Lumigraph [36] and the Light Field [51]. They considered only the set of light rays leaving (if we observe from outside) a convex bound of the examined scene so that the radiance along each ray remains constant. They also assume a snapshot of the function to eliminate time and considered a monochromatic function to avoid the need for examination over different wavelengths.

Both approaches parameterise the light rays by their intersection with two apriori known oriented planes. Synthesising an image from a novel view subsequently involves computing the four line parameters for each image ray and resampling the radiance at those line parameters. Gotler et al [36] approximated the resampling process as the linear sum of the product between a quadralinear basis function and the value at each grid point in the two planes. They also use prior knowledge of the geometry of the object to adaptively define the shape of this basis function. Levoy et al [51] interpolated the 4D function from the nearest point on the grid.

The simplicity of generating arbitrary new images once the approximation of the plenoptic function has been computed is the principal advantage of these methods. The pre-process required for approximating this function however is highly expensive in both computational and storage terms as the sampling density must be high enough to avoid excessive blurriness. Besides the compression improvements proposed this approach is not feasible for approximating scenes with a wide range of views such as buildings due to the prohibitive storage costs.
2.1.2 View Interpolation

The problem of the large number of images required in the preprocessing stage of all the approaches that approximate the plenoptic function can be circumvented with the use of geometric image inferences. View interpolation techniques utilise information as pixel depth or image to image constrains such as the fundamental matrix and the trilinear tensors to reproject image pixels from a small number of reference images to a given viewpoint.

Chen et al. [14] represented the whole set of original images as a graph structure. Each node in the graph contained a source image, the camera parameters and apriori known corresponding range data. Arcs connecting adjacent nodes represent directional morph maps (optical flow) obtained from the dense correspondences. A similar approach has been proposed in [13] where instead, a stereo algorithm has been used to compute approximate depth maps.

To synthesise views in-between a pair of images the displacement vectors are linearly interpolated and the pixels in the reference images are moved by the interpolated vector to their destination. However, linear interpolation only yields a valid reprojection if the source and the new image planes are parallel.

To overcome this limitation Seitz et al. [80] proposed a three step algorithm. Initially they prewarp (rectify) the source images so that their image planes are aligned. By linearly interpolating positions and colours on the reference images a new intermediate view along the line segment connecting the two camera centres is generated. A postwarping process finally transforms the image plane of the new view to its desired position and orientation. The principal restriction on this technique rely on the configuration of the two basis views which must satisfy the monotonicity constraint.

Generation of new views from rectilinear source images has been extended to cylindrical panoramic images. McMillan et al. [63] reconstruct panoramic views by stitching together images acquired from a purely rotated camera and describe a geometric constraint for cylindrical projections that determines the possible positions of a point given its location in some other cylinder. This constraint has been used to establish correspondences between cylindrical reference pairs which gives a dense disparity map. A
warp function subsequently combines the transformation of the disparity values from the known reference pair to the new cylinder and its reprojection as a planar image for viewing. Similar work using stereo pairs of planar images has been proposed in [48].

The introduction of the cylindrical epipolar geometry was the novel aspect of [63]. However, a more stable geometric constraint than the epipolar geometry is the trilinear tensor [83]. The use of trilinearities as a warping function from model views to novel synthesised images has been presented in [3]. A seed tensor is computed from three reference images. For every new view with known camera motion parameters relative to one of the reference images, a tensor is computed between the remaining two images and this new view. The tensor is subsequently used to render the image.

The relatively small number of reference images required by view interpolation together with the absence of explicit geometry are their principal characteristics. However, all such techniques rely on establishing correspondence between image pixels in the source views. Although, several geometric constraints have been imposed to solve this problem, in practice accuracy depends on the system calibration which is itself a challenging problem. Furthermore, when stereo correspondence estimation algorithms are used the small base line requirement, imposes limitations on the reference image configuration while the aperture problem limits the applicability of correlation techniques.

2.2 Geometry Modelling

Using photographs as the underlying scene primitives provides an inexpensive and realistic environment representation. However, if goals other than pure visualisation are important such as extensive walkthroughs in large scale sites and spatial reasoning or editing then an underlying geometric model that can be directly rendered from arbitrary views is required.

The problem of automatic reconstruction of 3D models has attracted considerable research interest, initially focused on modelling small objects and more recently larger scale environments. The form of these models can be polygonal, volumetric, based on higher order surface patches or CAD models. The geometric representation used de-
depends on both the type of acquisition sensor and the required use of the resulted model. Photorealism is enhanced by direct mapping of the images of the scene as texture onto the corresponding geometric primitives.

Geometry modelling methodologies can be categorised according to the amount of 3D information available and the type of the adopted sensors. Both active and passive sensors have been utilised to provide dense or sparse 3D measurements of the real scenes geometry. Unlike image based techniques, as the level of detail for a scene can be unrestrictedly complex, the computational requirements of the reconstruction process increases because of the bigger and more accurate data sets needed. Despite this principal limitation though, examples of reconstructions on complex real world environment have been achieved and constant advances in both sensor and computer hardware technology are likely to provide the means for arbitrary scene modelling in the future.

An overview of several different geometry based methodologies is presented in this section starting with techniques that utilise dense depth maps obtained from active sensors. Approaches that use the same kind of input data but instead based on camera sensors are subsequently presented. Using again passive sensing two categories of volumetric methods, shape from silhouettes and reconstruction based on the photo consistency constraint are used. Finally research has also addressed modelling from sparse data which is more closely related to this work. Despite the focus being on reconstructing large scale scenes, the feasibility of techniques that extend existing small object modelling approaches is also assessed.

2.2.1 Reconstruction using active sensors

Active sensing is more recent than passive however the associated technology has significantly progressed over the last decade making the use of such sensors more widely spread. There are two main categories of active sensors. The ones based on structured light triangulation and the laser range scanners (LRS). Sensors belonging in the first class project a light stripe on the scene and use a camera to view it. Based on accurate knowledge on the configuration of the light emitter relative to the camera depth can
then be computed. The second category of sensors emit and receive a laser beam and by measuring difference of phase, time of flight or frequency shift depth is measured. These latter sensors are the most widely used in robotics applications due mainly to their ability to operate on medium and long ranges.

The idea of using a laser range scanner for acquisition of 3D data is not new. Nitzan et al. [69] used a laser which was based on the phase difference between the emitted and the received beam in order to estimate distance to a point. At the time, the acquisition rate was 500ms per pixel making the formation of a 128x128 image a process lasting more than two hours. Now, it is possible to capture higher resolution (500x500) range images at a rate of several Hz. Despite the technological advances since then though, range sensors are not as wide spread as one might expected. The main reason is that the requirements for better speed and accuracy led to systems that were too bulky and costly for use beyond the research and industrial markets.

LRS have been used to obtain information required in the reconstruction of 3D representations for real objects and scenes. The reason is that such sensors provide an accurate range map of the scene in their field of view (FOV). In general however no complete model can be build from a single viewpoint due to the occlusions caused by the complexity of the scene surfaces and the limited sensor’s FOV relative to the size of the scene to be reconstructed.

By deploying the LRS in different positions potentially the whole scene can be scanned. However, this acquisition approach yields to the important problem of registering the partially overlapping images in a common coordinate frame. The goal of registration is to find the transformation that best aligns the range images. There are two main methodologies addressing image registration. The first is to practically circumvent the problem completely by relying on precisely calibrated mechanical equipment such as turntables and eye-in-hand devices (robot arms) to determine the motion between the views. The second consists of methods that exploit information contained in the overlapping range images.

The vast majority of registration techniques are based on the Iterative Closest Point (ICP) method. The main idea is to refine a rigid transformation by iteratively com-
puting the incremental transformations that minimise the distance between the transformed points from the first view to the second. The first developed and most well known algorithm based on this strategy has been proposed by Besl et al. [8]. They assumed correspondence between points based on the closest distance metric and compute the rotation and translation vectors that minimises the residuals at each step in a least squares sense. The original ICP algorithm has several limitations the most prominent of which are that a good initial estimate of the transformation is required in order to ensure convergence, the method is not robust to noise and one data set is assumed to be a subset of the other. Further research has successfully addressed all these limitations.

Once the multiple images are registered into a common coordinate frame range data from regions of the scanned surface captured from more than one image can be integrated to obtain a single fused surface. Strategies for integrating multiple range images can be classified to mesh and volume based according to the intermediate representation used. Mesh integration techniques [89, 98, 76] initially perform a triangulation to the raw 3D data, identify the overlapping regions, remove the redundant geometry for both meshes and in a final step stitch them to a single representation. Although these methods appear to perform better in computational terms [40] they can still fail catastrophically in areas of high curvature [20].

Volumetric based integration methods [41, 20, 74] are generally more robust. The whole scene volume is subdivided into a discrete set of voxels and for each image a continuous function that represents the weight signed distance of a point to the nearest estimated surface is sampled at the vertices of a voxel. Voxels that cover regions of the real surface which have been scanned several times will have multiple estimates of the function on each vertex which are combined according to their weights. The isosurface that corresponds to the zero set of the function is considered as the reconstructed surface. The calculation of the zero points of the function and its polygonisation can be achieved by the marching cube algorithm[53].

The methodology involving acquisition, registration and integration of multiple scans has been extensively applied to the reconstruction of small objects. and applications
have been successfully developed to cover the market demand. Inherently though this reconstruction approach cannot be directly extended to build 3D models for large scale scenes because of their computational complexity and storage costs. Image registration using ICP is a iterative minimisation process over hundreds or even thousands of points while image integration can be shown [40] to exhibit a $O(M^2N)$ computational and $O(N^3)$ storage complexity where $M$ is the number of images and $N$ the average number of points in each range image.

An attempt to develop a system for scene reconstruction based on this methodology has been presented by El-Hakim et al. [29]. Their system consists of eight CCD cameras and a LRS mounted on a mobile platform. For each position of the platform the LRS produces a range map for each of the 8 images. Features corresponding to discontinuities in both images and range maps are automatically extracted but manually matched. Bundle adjustment is used to compute the positional parameters of each image in a global coordinate system. Correspondences between 2D and 3D features are then used to register the range data which are subsequently integrated by a volumetric method [74]. Although no direct indication for the execution time of the reconstruction process has been presented the resulting model for a single empty room consisted of several thousand triangles for the coarser voxel resolution.

The size of the reconstructed models is directly related to the amount of data yielded by each scan. If this raw data is triangulated into facets then an extremely large and complex mesh representation is recovered for the scene. To remedy this inherent problem Sequeira et al. [81] tried to automatically fit planes to the 3D measurements. A nearest neighbour multiresolution triangulation guided by the surface information is subsequently used to yield a triangular mesh. The registration is performed by an ICP algorithm while a mesh based integration process that disregards the less reliable points on overlapping regions yields the final model. The method is recursive and their mobile platform is driven by a view point planning algorithm that detects occlusions based on surface discontinuities and selects the next capture point that optimises a number of criteria such as occlusions resolution and acquisition conditions. A similar but batch approach has been presented by Stamos et al. [90].
More recently El-Hakim et al. [28] have extended the plane fitting method to more general surface patches and CAD models. However, their method is highly interactive as manual intervention is necessary in both the registration stage and the feature grouping where the human operator is required to draw with the mouse a window bounding the points that belong to the same surface. Each of the segmented patches is subsequently triangulated and textured mapped separately which results in intersecting surfaces not connected in the model.

Plane fitting is a good method to reduce the amount of data acquired by the laser scanner. Nevertheless distinguishing scene features smaller than the threshold adopted for considering data as part of a surface becomes unfeasible. Data corresponding to these features collapse to their assigned surface and so disappear from the reconstructed model. Similar effects characterise an alternative data reduction approach which is based on mesh simplification as triangles collapses based on measures such as curvature or locality. A reconstruction method that uses such a reduction approach has been proposed by Hubert et al. [42]. For each range scan they compute a simplified mesh and measure the local surface shape at specific points using spin images which is a 2D signature. Measuring similarity between these spin images they estimate an initial alignment which subsequently improved with an ICP algorithm.

Data reduction is essential for reconstruction of small, flexible and versatile models however it cannot be applied to modelling methods [9, 75] aiming at recursively “carving” a solid volume around the scene. The main idea behind these methods is to subdivide the space into voxels that are initially considered as full, deploy the LRS somewhere in this solid environment and emit a series of beams within its FOV. Any space between the scanner and each point on the surface of the scene pointed by a laser beam is considered empty (Figure 2.1). By repositioning the LRS and acquiring new range maps a model of the occupied space is iteratively built. If this process is repeated by selecting new views for regions of space not previously scanned the approach can converge to a full model of the environment.

Volumetric modelling techniques like [9, 75, 29] although applied in reconstruction of large scenes always result in huge impractical polygonised models. Although the size of
2.2. Geometry Modelling

Figure 2.1: Model reconstruction by recursively carving a solid volume

such models can be significantly reduced by decreasing the voxel resolution this appears limited as the size of the voxels sets the bound on the expected error so that if it exceeds a certain threshold the advantages for using a LRS are disregarded.

2.2.2 Reconstruction from dense passive sensors range data

Stereo vision is the process of acquiring 3D range information about a scene from two or more images taken from different viewpoints. This is similar to the human visual system where the different perspectives of our two eyes result in a slight displacement of the scene in each of the two monocular views that permits us to estimate depth.

Computer stereo vision is a passive sensing method which is based on triangulation between the pixels that corresponds to the same scene structure projection on each of the images.

Two views are sufficient in order to compute 3D depth information. However, this does not always correspond to the number of physical cameras. Triangulation can be achieved using a single camera that moves around the scene by considering its previous captured image as its stereo pair. A system that uses a single uncalibrated hand
Chapter 2. Overview of related work on Scene Modelling

A held camera for full scene reconstruction has been proposed by Pollefeys et al. [71]. Initially they compute the epipolar geometry that relates the first couple of images in the sequence and define a reference frame based on it. The camera pose for each of the subsequent frames is then estimated in this reference projective frame. Reconstruction using uncalibrated stereo can be only determined up to a projective transformation and thus the next step involves restriction of this reconstruction ambiguity to a metric one. Having calculated the relative position and orientation for all cameras, image rectification is applied in order to facilitate a dense depth estimation process. The raw 3D data are subsequently smoothed using a thin plate spline method. In the final step a triangulation in the reference view is applied in order to build a piecewise surface model. The principal weakness of this system is the lack of integration of data from different viewpoints which limits the reconstruction to selected regions of the scene.

The use of a single camera imposes the constraint of a short baseline between the successive views in order to achieve accurate automatic matchings. On the other hand, this results in extraction of relatively inaccurate depth data located within a narrow field of view. A system that overcomes this problem has been presented by Kang et al. [43] and make use of panoramic images taken from a single weakly calibrated camera. At each camera position a panorama is created by stitching images captured while rotating the camera 360° about its centre. Extraction of features and matching among the panoramic images is used as input to an eight point algorithm from which the essential matrix between a reference panorama and all other images is computed. Using a constrained search based on the epipolar geometry dense matching is achieved that results in a dense depth map. By applying a 3D triangulation on the cloud of estimated points a model of the real scene is computed. Unfortunately, this technique is not applicable for indoor environment reconstructions because of the inherent limitation in modelling the floor and ceiling of the scene.

Contrary to [71] and [43] where monocular video has been used Narayanan et al. [67] employed an acquisition method based on 51 synchronised calibrated cameras mounted on a 5 meter diameter dome. Their “virtualised reality” system gave the user the ability to control the viewing angle of a dynamic event taking place in the dome. At each time instant a multi-base line stereo algorithm is utilised to compute a dense
range map. Nearest neighbour triangulation is used to build a polygonised model for each viewpoint. Subsequently the model is texture mapped from the corresponding source images providing the ability to directly render the view from each of the original camera’s directions. To achieve view synthesis from an arbitrary viewpoint the model from the nearest camera was chosen as reference and rendered. Holes due to regions that were originally occluded in the reference frame and become visible from the new viewing direction are covered by rendering neighbour models. Alternatively the system could provide a complete surface model by fusion of the multiple depth maps using a volumetric integration method [20]. Although both dynamic and static scenes can be modelled the hardware configuration of the system is both costly and static. Only events inside the dome can be modelled making the approach infeasible for applications where cameras should be deployed inside the scene for exploration and mapping.

The methodologies for reconstruction from dense depth data acquired from LRS and stereo images are very similar as they both rely on registration and integration steps. Calibration, which can be considered equivalent to registration, in stereo techniques is a much simpler process which usually can be performed off line. Nevertheless stereo vision involves a step consisting the correspondence and triangulation processes in order to estimate depth. Achieving reliable, robust and accurate automatic correspondence between multiple views of an arbitrary scene though, is still an open problem in computer vision. As this process of estimating depth is performed by hardware using a LRS, superior range resolution and accuracy can be achieved. Despite the cost, size and time this is the main reason that active scanners are still utilised for applications of scene reconstruction where accuracy is the principal requirement.

2.2.3 Shape from Silhouettes

The earliest attempts in reconstruction of 3D models from photos used the silhouettes of objects as sources of shape information. A 2D silhouette is the set of close contours that outline the projection of the object onto the image plane. Segmentation of the silhouettes from the rest of the image and combination with silhouettes taken from different views provide a strong cue for image understanding.
Typically shape from silhouettes techniques start with an acquisition step where images of the object are taken from different locations around it. For each of these images the object silhouette is extracted using simple differencing or blue screen segmentation techniques. The computed silhouettes for every image along with the centre of the corresponding camera is then used to define a volume which if backprojected to 3D space can be assumed to bound the object. The intersection of these volumes associated with the set of acquired images yields a reasonable approximation of the real object. This intersection volume has been named the visual hull by Laurentini et. al. [46] and described as the maximal object that gives the same silhouette with the real object from any possible viewpoint.

A property of the visual hull is that as the number of images used increases, its fit to the actual object volume becomes tighter. However, this number can be proved [47] to be unbound for reconstruction of general polyhedral objects. Even if the acquisition of an infinite number of images was possible, silhouettes can be insufficient clues for fully compute the shape of non convex objects. An example where the silhouette methodology will fail is illustrated in figure 2.2. The reason for this limitation is that concavities in the object geometry result in self occluded areas for the object that cannot be resolved from any viewpoint unless additional information is provided. Niem et al. [68] and Szeliski [94] have used turntables and calibrated cameras to produce 3D range information for pixels belonging to the object silhouette which subsequently
2.2. Geometry Modelling

integrated with the volume intersection model to correctly estimate the object’s shape.

Unfortunately, the type of the object is not the only parameter affecting its corresponding visual hull form. The positioning of the cameras can significantly influence the computed model especially when the number of acquisition locations is small. An iterative method to specify the viewpoints that optimise the reconstruction taking into account some prior knowledge on the object shape has been presented in [82].

Shape from silhouettes is a particularly good approach if only a crude model of the real world is required. The methodology is intuitive and easy to implement and this is the main reason that systems generating and replaying 3D digital video [64] as well as commercial object modelling packages [68] are based on it. Nevertheless, reconstruction is restricted to small solid objects for which their whole geometry can be captured from photos around them and thus are not applicable to scene modelling.

2.2.4 Reconstruction based on photo consistency

Colour or greyscale variance provides information that is not exploited by shape from silhouettes methods that only use segmented binary images. This information can be considered as constraints in the 3D reconstruction as a valid point on the scene surface appears with the “same” colour over all images that are visible, under the assumption of constant illumination and Lambertian reflectance. This constraint is known as photo consistency and is the basis of a whole category of techniques on volumetric reconstruction.

Seitz et al. [79] first and Kutulakos [45] more recently have proposed a methodology for reconstructing scene models based on photo consistency. Initially a volume that encompasses the whole real world scene is defined and subdivided into voxels. The set of voxels is assumed to be either transparent and subsequently labelled as opaque if they pass the photo consistency test [79] or assumed solid and become transparent if they fail the test [45]. Testing photo consistency involves projecting every voxel centroid onto each of the images from which it is visible and thresholding the variance of the colours of the associated pixels in these images. This check however, requires accurate camera calibration and is very sensitive to noise.
Unfortunately, the photo consistency constraint does not guarantee a unique reconstruction as a whole set of models maybe consistent to the input images. To circumvent the ill posed nature of the problem, [45] has presented an algorithm called “space carving” which yields a unique reconstruction called the photo hull. This is the union of all photo consistent subsets of the scene. In this sense the photo hull is not a minimal reconstruction but one which ensures that no valid voxels are disregarded from the consistency test.

This test has to be performed for every voxel based on its projection on all the images from where it can be visible. This can be a very computationally expensive process as elimination of a voxel automatically results in changes of visibility for its surrounding voxels. To mitigate this practical limitation, [79] has proposed the ordinal visibility constraint where no scene points are contained inside the convex hull formed by the positions of the cameras. In such a configuration the visibility of every two points relative to a camera can be decided based on the distance of the points from the hull. Placing the cameras in a plane facing to the real scene permits a partitioning the 3D space into layers of uniform depth from this plane. Traversal of voxels in layers of increasing depth guarantees that before any voxel is visited all its possible occluders have been checked.

This approach has the advantage that each voxel is only visited once. However, the ordinal visibility constraint is very restrictive on the employment of the cameras as it is not possible to surround the scene with cameras. This problem has been addressed in [45] by a multi plane sweep technique along x,y,z coordinates and in both positive and negative directions. Evaluation of the consistency test is performed for each voxel in the specific plane using only the subset of cameras that lie infront of the plane.

Although this approach places no restrictions to the locations of the cameras relative to scene, checking the colour consistency of a voxel only uses a subset of images from which the voxel is visible. An alternative method has been proposed by [19] called generalised voxel colouring (GVC) where a layered depth image (LDI) data structure has been utilised so that every pixel is associated with a linked list of voxels that project over it sorted in depth order. Photo consistency is tested iteratively on every surface.
voxel (head of LDI), successively carving invalid voxels and replacing them with the subsequent voxels in the corresponding LDI entry until no change occurs in a complete pass over the surface. Although GVC achieves faster convergence over the multi plane sweep methodology it is applicable to cases where cameras are placed outside the voxel space.

Despite this limitation Slabaugh et al. [87] addressed the problem of large scale scene reconstruction using GVC. They subdivided the voxel space into an interior space where reconstruction of foreground surfaces is performed and an outer space which is used to model the background scene. This exterior space is warped so that the size of the voxels increases proportionally to their distance from the interior space with voxels on the shell of the exterior space having coordinates to infinity. To overcome the problem appeared from the cameras get embedded inside the voxel space since the exterior volume covers the whole scene, they manually precarve the volume. An algorithm similar to [19] was proposed for carving with the difference that voxels on the shell of the exterior space never become invalid irrespective of the photo consistency test as one cannot see beyond infinity. This approach yields improved results for outdoors scenes as both the foreground and the background are modelled. However it is not applicable to large scale indoor environments where visibility is restricted from wall structures and a large number of cameras have to be deployed making the precarving process infeasible.

Building a model for a large scale scene using volumetric representations is a challenging problem as the number of voxels required to bound the scene become prohibitive to process. A multi-resolution approach has been presented by Prock et al. [72] that attempts to tackle this problem using a coarse to fine strategy. They proposed covering the initial voxels space with low resolution voxels (large size) and testing photo consistency using the mean colour over the set of pixels to which a voxel is projected. The resulting model contains a large number of gaps and missing voxels as the lower resolution voxels may encompass only a small part of the real scene. By using a nearest neighbour search, they add back for each of the remaining colour voxels its four adjacent voxels, subdivide them into eight and test consistency for each of these new voxels. This approach appears to fill the voxels that were deleted as false negatives and considerably reduces the overall number of voxels used to reconstruct the scene.
Chapter 2. Overview of related work on Scene Modelling

The advantage of reducing the number of voxels is essential especially when objects further away from the camera have to be modelled. Kutulakos [44] presented another fine to coarse approach that enable such reconstruction by searching photo consistency in a circle of specific radius around the pixel that one voxel projects in each image. He called this test r-consistency and shows that the size of the radius can be considered as the controlling parameter to the detail of the reconstructed model. R-consistency yields a hierarchy of models where the photo hull is the finest and closest approximation to the real scene.

The relaxation of the photo consistency definition proposed in [44] results in robust reconstruction under calibration and image noise. Another approach that appears invariant to noise has been described in [27] where a multi hypothesis voxel coloring has been adopted. In an initial step they perform hypothesis generation for each voxel by testing photo consistency over every pair of views. If the voxel has the same colour in at least two cameras a new hypothesis is assigned to it. In the subsequent hypothesis removal step for each view the currently visible voxels are determined and their associated hypotheses are checked for consistency with the colour of the pixel they project onto. A voxel is carved from the voxel space if all its hypothesis are invalid. This approach has the advantage that carving is performed one image at a time and so voxels are visited less times. However, in the preprocessing stage even the intermediate voxels have to be coloured which adds a significant overhead to the methodology.

The constant illumination assumption adopted by all the voxel colouring methods can be relaxed in a multi hypothesis approach as only two images are enough to generate a valid voxel colour estimation. However, the method is still susceptible to calibration errors as only the centroid of each voxel is projected to each image and gets tested. Calibration is a difficult problem and becomes even more crucial when more than one cameras are used to capture the scene geometry. Saito et al. [78] presented a method that requires only the fundamental matrices that relate every image. In their scheme they select two basis views and define a projective voxel grid space. All voxels can then be reprojected onto any arbitrary view knowing the fundamental matrices that relates this view to the two basis images. This approach however has the disadvantage that the choice of the basis views affects the reconstruction quality as the shape and
size of the voxels is determined by their location relative to the cameras that form this projective basis.

2.3 Modelling from sparse data

Passive capture compared with active sensing exhibits some significant advantages in terms of cost, practicality, complexity of use and integration to a larger system. These are the characteristics that make cameras the more attractive sensor for a wide market reconstruction product. Unfortunately unless accurate calibration and precise information on the pose of the stereo cameras are known, 3D measurements can only recovered for a sparse set of features. As these features correspond to discontinuities in the intensity of the images and are usually associated to “significant” structure in the scene they can be reliably matched and tracked along successive frames.

Reconstructing a 3D model of a real environment involves both estimating its geometry and topology. Scene geometry refers to the 3D position of the recovered structure primitives while scene topology refers to the set of surfaces that connect these primitives together. Although significant work has been done towards the first direction, only limited attempts has been made to recover topology.

2.3.1 Structure from motion

Automatic feature extraction and matching are probably the most well studied problems in computer vision research field. Once the 2D projection of a point in the real scene has been found, its position in 3D can be assumed somewhere along the ray connecting the camera optical centre and the corresponding spot in the image plane. Tracking its projections across multiple images and using triangulation [39] allows the relatively accurate localisation of the point in 3D. If extraction and corresponding can be performed for a sufficient number of points and lines and over images acquired from different directions then estimates for both the 3D locations of the features and the camera positions can be deduced. The obtained reconstruction however differs from the true structure by a projective transformation. Knowledge of camera intrinsic para-
meters reduces this ambiguity to metric. This reconstruction method is called structure from motion (SFM) and is based on the bundle adjustment process [97] of minimising the distances between estimated 3D structure projections and actual image measurements. An extensive literature exists for SFM but covering this is out of the scope of this study. Nevertheless a good review of this methodology can be found in [31].

Earlier attempts like the DROID 3D system and the Vanguard project aim to build a 3D representation of a site by using passive sensors and utilise SFM to obtain 3D estimates of the scene structure. Unfortunately, both focused on the approximation of the scene geometry and fail to address the problem of reconstructing the surfaces that connect the available cloud of 3D features. Solutions like off line fitting of surfaces [5, 33] resulted in simplified and unconnected models while attempts to build a continuous triangulated model by a single view planar triangulation method [38] are only valid in the absence of occlusions.

2.3.2 Single view reconstruction

Inferences about the 3D structure can be made from a single uncalibrated camera if vanishing points and lines can be computed. Assuming a simple camera model with zero skew and aspect ratio of one and using three orthogonal pairs of parallel lines for recovering three of the remaining internal camera parameters a weak calibration can be performed [12]. Once calibration is achieved a single vanishing line allow complete rectification of the corresponding plane and determination of its orientation relative to the camera. Using this methodology however only 3D directions for lines and planes can be estimated. Nevertheless automatic identification of perpendicular sets of lines is very difficult as projective transformations do not preserve angles.

Several interactive methods that circumvent these limitation by using information provided by the system user have been presented. Liebowitz et. al. [52] initially choose a reference plane and compute its vanishing line and a vanishing point for its orthogonal direction using sets of manually supplied parallel lines. By specifying the relative distance of a single point from the reference plane the orthogonal distance of every point in the image to that plane can then be computed. Metric rectification of the reference
plane allows absolute 3D positioning of points in space and feature grouping yields a simple planar scene approximation. Similar results has been presented in [92]. However, in that approach reconstruction has been formulated as a minimisation process over the sum of squared distances for pairs of planes and points or lines. Originally this technique has been proposed by Shum et al. [86] and applied on panoramic images. They estimated camera rotation and plane normals and recovered plane distances, 3D structure and camera translation by solving a constrained least squares problem. Their system also extended to fuse multiple panoramic views using bundle adjustment.

2.3.3 Multiple view reconstruction

Reconstruction of 3D models from a single image can produce only limited simplified representations of the scene as it can be seen from the specific viewpoint. To resolve occlusion ambiguities multiple images that sufficiently cover the whole scene should be acquired. PhotoBuilder [73] is an interactive working application for modelling architectural buildings from video that can be considered as an extension of [86]. Unlike [86] though motion and structure estimates are computed using a batch bundle adjustment method and planes are determined by a best fit algorithm over their manually grouped features. The final model is a textured triangulated representation constrained to the computed structure primitives.

The problem of automatically associating individual features to higher level surfaces in 3D is very complicated because no knowledge of the spatial extent of these surface patches can be accurately inferred. The simple solution would be to use manual intervention to gain this knowledge. However, automatic plane reconstruction methods have been presented [4, 23]. Bailard et al. [4] have proposed a batch approach where 3D line positions are estimated by tracking their projections over successive frames and a plane sweep strategy is then applied by hypothesising a planar facet attached to each line for every angle around it. The half plane with the highest similarity correlation score over the whole set of images is assigned to the line and a set of heuristic rules are subsequently used to support line grouping and outline the boundaries for each estimated plane. The approach is very computationally expensive while the heuristics
used may fail under complex planar configurations. A more efficient algorithm optimised for reconstruction of architectural scenes has been presented in [23]. Estimates of 3D points are computed and an initial plane grouping is achieved by recursively applying a RANSAC plane estimation algorithm. All structure is then projected to the ground plane. As all other planes corresponds to walls that can be considered perpendicular to ground they fit line segments to these projected 2D points and hypothesise the plane boundaries from the highest and lower height points in these clusters. This process unfortunately is only applicable to a restricted number of scenes as it can only recover the ground and the side planes of buildings. Nonetheless the resulting model is not a coarse planar approximation as a region growing algorithm is proposed to search for areas within each plane whose likelihood is maximised at similar offset values. The position and size of these offsets is manually fed to the system and their shape is automatically decided from a set of predefined models.

The idea of incorporating geometric models common to architectural buildings has previously been presented byDebevec et al. [21]. They developed Facade, an interactive tool for reconstruction of realistic models from a sparse set of images. Initially the user builds a hierarchical geometric model that closely resembles the scene. The model consists of predefined parametric polyhedral primitives and the process used is similar to solid modelling. A reference image is chosen and an estimate of the corresponding camera rotation and translation relative to this models coordinate system is computed. Model parameters are optimised by minimisation over the disparity between interactively matched projected edges of the model and edges marked in the image. Although the resulting model accurately approximate the geometry and topology of the scene additional detail may be captured using a dense model stereo method [22]. Although the approach requires significant manual intervention it combines geometric and photogrammetry techniques resulting in highly realistic reconstructions.

A method aiming at scene topology estimation assuming only a sparse set of 3D point estimates computed from a SFM process has been proposed by Morris et al. [65]. A triangulation of this point set is initially performed and the resulted model is then projected onto each of the images in the sequence. Using successive edge swapping and measuring the variance of the intensity for pixels belonging to corresponding triangles,
they formulate the problem of finding a consistent triangulation as a minimum variance estimation problem over the space of all possible triangulations. Although the method seems to produce good surface approximations for complex real scenes its principal disadvantage is that no occlusions can be handled.

A system that attempted to address both problems of estimating scenes geometry and topology automatically has been developed by Faugeras et al. [32]. They use a trinocular video head mounted on a mobile robot that moved around the scene trying to build a 3D model representation of the real world environment. For each position a local 3D geometry description is computed called the “Visual Map” based on the stereo reconstruction. By relying on a batch Extended Kalman Filtering approach they achieve registration and fusion of these local maps into a common global coordinate system. Scene topology is estimated off line after the whole sequence has been processed by applying a Constrained 3D Delaunay triangulation over the reconstructed 3D structured primitives. This process yields a volumetric triangular face representation difficult to texture or render. For each viewpoint, the visibility of scene features from the corresponding “Visual Map” is tested in order to remove parts of the model that occlude the features. This is a process that requires accumulation of all the “Visual Maps” during the reconstruction stage. Furthermore, no use of uncertainty over the feature positions has been utilised in the visibility tests and therefore the method is highly susceptible to noise as only small perturbations in feature localisation can result in rejection of valid scene regions. Finally the process is strongly dependent on the locality heuristic as the generation of the tetrahedral hypothesis is performed in the single global coordinate frame. Removal of estimated tetrahedra according to visibility constraint from previous viewpoints can thus yield holes in the model without any chance of filling the resulted gaps.

More recently an incremental geometric theory for estimating the scene topology has been presented by Manessis et al. [57] and it has been proved that it converges to a triangular surface approximation of the real scene as the number of views increases. An algorithm has also been introduced that gives a computationally efficient approximation of the general methodology where a consistent model is progressively built without the use of history information stored from previous frames. Robustness of the
reconstruction process to outliers and noisy input 3D measurements has further been presented in [55]. This is a complete planar scene topology estimation system operating on sparse input data that addresses all the problems presented in [32].

2.4 Reconstructing representations for real world scenes

Reconstruction of real world scenes and smaller scale objects has received a great research interest in recent years driven by the numerous applications in virtual and augmented reality systems. A system that would provide a general solution to this problem should satisfy several criteria:

- **Automatic** to limit the reconstruction time and requirement for expert technical knowledge.
- **Low-cost** to address mass-user applications.
- **Accurate** to correctly approximate the scene geometry and topology.
- **Reliable** in the absence of distinct features or surface texture and presence of changes in appearance due to illumination and surface properties.
- **Photo-realistic** to synthesise a walk-through of an equivalent visual quality to a real video sequence.
- **Versatile** to reconstruct multiple scale scenes and objects.
- **Compact** to produce models suitable for transmission and rendering.
- **Computationally Efficient** to allow simultaneous image capture and model reconstruction.
- **Incremental** so that the user can identify occluded regions and capture additional images to reconstruct the missing parts of the model.

Unfortunately, no single system currently fulfils all these characteristics and different approaches based on different sensors are currently adopted according to the application priorities. When accuracy is the predominant requirement over cost, compactness
and efficiency laser range scanners are the preferred sensor solution. Relative to LRS, reconstruction from photos has also the substantial drawback of being much more sensitive to environmental conditions such as illumination and scene appearance. However, today’s digital cameras provide fast and high resolution imaging at affordable rates and so photos can be the favourite input primitive to either image based methods for simple visualisation and navigation applications or geometry based methods for 3D mapping, extensive walkthroughs or video editing.
Chapter 2. Overview of related work on Scene Modelling
Chapter 3

Theory of reconstruction from sparse data

This chapter presents a provable geometric theory for reconstruction of surface models from sparse 3D data captured from N camera views. The location of features on the surface of objects in the 3D scene, known as scene geometry, is assumed to be computed by an automatic recursive SFM algorithm. The theory is independent of the SFM algorithm adopted and based on its output 3D structure and camera motion estimates it guarantees a reconstruction of the scenes topology which is consistent with the feature visibility using a hypothesise and verify approach.

No prior assumptions are made about our knowledge of the structure of the 3D scene, scene illumination, surface properties or the ability to separate the foreground and background. The methodology presented in this chapter considers the general problem of reconstruction from sparse feature data in the absence of noise. Noise in the input measurements is a problem which is addressed in chapter 6.

3.1 Problem statement and definition of a consistent model

For N-views of a 3D scene we assume that we can reconstruct estimates of the 3D location of a set of scene features and the camera parameters for each view by correspondences over successive frames. Features may include points, junctions, lines, curves
and generally any image structure that can be matched across multiple images. The feature set may be considered either sparse or dense depending on whether local surface topology can be estimated directly from the Euclidean distance between them.

From topology [18] we can assume a mapping function $f : X \to Y$ between domains $X, Y$ to be continuous at $x$, where $x \in X$, if for each neighbourhood of $f(x)$ in $Y$ there is a neighbourhood of $x$ in $X$ whose image under $f$ lies in the neighbourhood of $f(x)$ under consideration. If we define the neighbourhood (of $x$) to be the set of all points (in $X$) with distance less than a prespecified radius ($\delta$) and denote it as $N(x, \delta, X)$ then the continuity requirement can be restated as:

$$f(N(x, \delta, X)) \subset N(f(x), \epsilon, Y)$$  \hspace{1cm} (3.1)

where $\epsilon$ defines the neighbourhood in $Y$. We shall say that $f$ is continuous if it is continuous at every point in $X$.

For dense data if the Euclidean distance between features is less-than a fixed threshold it may be reasonable to assume that the surface is at least geometrically $G^0$ continuous in position between them. A dense range image can be considered as a function which associates 3D surface points to a $2 \frac{1}{2} D$ image pixel. For every neighbourhood of four points laying on adjacent rows and columns of the image, real scene surface continuity may be assumed by examining the corresponding mapped surface points. Topology can then be estimated by hypothesising zero, one or two triangles that connect triples of these adjacent image points where the function is continuous [98, 89, 29, 81, 67]

For a sparse set of features the surface topology can not be estimated directly from the relative location of the extracted scene primitives. The reconstruction problem to be solved in this case can be stated as:

Given a set of sparse 3D features captured from N-views of an arbitrary unknown 3D scene, reconstruct a surface model that consistently approximates the scene surface topology.

where consistency is considered according to the following Definition 2.
A recursive SFM reconstruction algorithm computes a set of 3D feature estimates $F_i = \{f_j\}_{j=0}^{N_f}$, and camera parameter estimates $P_i$ for each view $i$, together with estimates of their uncertainty. Each feature $f_j \in F_i$ represents the portion of the feature visible in frame $i$. The set of the greatest extended parts of all features that have become visible is assumed to be $F$.

Every set of sparse 3D features $F_i \in F$ viewed from a single camera with location $\vec{v}_i$ provides information not only on the spatial position of the real-scene features but the empty space between the visible part of the feature and the camera. Thus each feature $f_j$ visible in view $i$ defines a visibility constraint, $c_{ij}(f_j, v_i)$, defined as follows:

**Definition 1 (Visibility Constraint):** The space between the view position $\vec{v}_i$ and the visible part of the scene feature $f_j \in F_i$ is not occupied by an (opaque) object.

This constraint for a point feature, $\vec{x}_j \in \mathbb{R}^3$, indicates an empty ray in space between the camera location, $\vec{v}_i \in \mathbb{R}^3$ and feature location, $\vec{x}_j$. For the visible part of a line feature represented by its endpoints ($\vec{x}_1, \vec{x}_2$) the visibility constraint defines an empty triangle in space ($\vec{v}_i, \vec{x}_1, \vec{x}_2$). Likewise for the visible part of an arbitrary curve feature the visibility constraint is an empty ruled surface. These surfaces are composed of the set of lines passing through the camera location $\vec{v}_i$ and consecutive control points on the curve. For the remainder of this thesis it will be assumed that curve features are approximated by a set of line segments defined by control points on the curve.

The reconstruction problem has been defined as the computation of a ‘consistent’ 3D model which approximates the scene topology given a set of 3D features, camera parameters and feature visibility constraints according to Definition 1. A consistent representation is defined as:

**Definition 2 (Consistent Representation):** A consistent 3D model is a set of surfaces which interpolate the space between the sparse 3D features, $F$, and do not violate any of the feature visibility constraints, $C$. 
Figure 3.1 illustrates consistent and inconsistent representations according to Definition 2 for a simple scene containing three line features. In figure 3.1(a) the scene representation intersects the visibility constraint for one of the line features resulting in an inconsistent model. In contrast, figure 3.1(b) shows one possible scene representation which is consistent with the triangular feature visibility constraint.
of dense data we may make the additional assumption about surface continuity from the distance between features which considerably simplifies the surface reconstruction problem. This definition can also be applied to scene reconstruction using active sensors where features in the form of direct 3D measurements are captured from N known sensor locations and feature visibility takes into account the sensor geometry.

3.2 Reconstruction of a Consistent 3D Scene Model

An arbitrary real 3D scene can be represented by a set of surfaces $S_i$ that span the space between the feature primitives that describe the scene. This set of surfaces can be approximated to arbitrary precision by a triangular mesh $S \approx M = \{t_i\}_{i=1}^{N_t}$. Triangulated surface models are the simplest representation and do not require any prior assumptions on surface type [7].

We approximate the topology of an arbitrary unknown scene by a constrained triangulation on the set of features $F$ such that points and lines are included in the model as triangle vertices and edges. Then for $M$ to be a consistent representation according to Definition 2 each triangle $t_i$ must satisfy the set of N-view visibility constraints, $C$. Thus we obtain the following definitions for a consistent triangulation:

**Definition 3 (Consistent Triangle):** A triangle $t_i \in M$ is consistent if it does not intersect any of the visibility constraints in $C$.

**Definition 4 (Consistent Triangulated Model):** A model $M$ is consistent if all triangles $t_i \in M$ are consistent according to definition 3.

There is a family of possible consistent triangulated models, $M^*$, which interpolate the scene features $F$ given the assumption that we can approximate the scene surfaces, $S$, by planar patches. This assumption is valid for arbitrary shaped scene surfaces provided that a set of scene features, $F$, can be reconstructed which approximate the scene geometry. In the remainder of this chapter we introduce an algorithm to reconstruct one such model, $M \in M^*$, which is guaranteed to be consistent with the N-view feature
visibility constraints, \( C \), and prove that this approach converges to a piecewise planar approximation of the true scene surface, \( S \), as the number of views increases.

### 3.3 Single-view Reconstruction

Every planar region defined by a set of points or a polygon can be partitioned to a set of triangulated facets. Such a space subdivision is defined as a closed planar triangulation if the following criteria are met:

1. Each pair of triangles are either disjoint or share a vertex or have two vertices and their associated edge in common.
2. There is a path along the edges that links every two vertices.
3. There are no holes.
4. For every vertex the set of edges opposite to it form a connected path.

When the input set is a planar straight line graph of points and non crossing (intersecting only at endpoint) line segments under the requirement that these feature primitives are preserved at the final mesh as vertices and edges correspondingly then the triangulation is called constrained.

For each view \( i \) we assume that a set of 3D visible features \( F_i = \{f_j\}_{j=0}^{N_i} \) has been computed and we use a 2D triangulation method to estimate the scene topology. The problem of spanning the space between the cloud of raw 3D features reduces to the construction of their closed constrained triangulation in a plane orthogonal to the camera view direction given by the projection, \( P_j \). Assuming that the scene is opaque, as planar segments corresponds to projections of real observed features and perspective imaging is injective (any pair of distinct picture elements are associated to two different world points) no planar crossings may occur except among features belonging to a textured surface. However such cases may identified by examination of the relative depth of the intersection point against the associated 3D lines and participating segments can be split.
Once the planar mesh $T$ has been constructed a surface triangulation is obtained by backprojecting $T$ to 3-space using the 3D estimates available for the feature positions. This produces a piecewise planar surface mesh $M_i$ in which the original extracted structure has been preserved. Furthermore, as the order of feature projections, $P_j(f_i)$, in the plane is maintained with respect to their relative ordering in real space from the camera viewpoint the following proposition holds:

**Proposition 1:** Constrained triangulation of the projected features, $P_i(F_i)$, in the 2D plane orthogonal to the view direction according to the camera projection, $P_i$, results in a model, $M_i$, which is consistent with the visibility constraints $C_i$.

**Proof:** Assume that the reconstructed 3D triangulated model is not consistent with $C_i$ and part of a visible feature $f_j$ lies behind one of the 3D triangles $t_i = (\vec{x}_1, \vec{x}_2, \vec{x}_3)$ of $M_i$. If we consider the 3D prism $pr_i = (\vec{v}_i, \vec{x}_1, \vec{x}_2, \vec{x}_3)$ defined by the camera centre $\vec{v}_i$, the corresponding image plane triangle and $t_i$, then in the case $f_i$ is a point it should lie inside $pr_i$ while if $f_i$ is a line segment it should either be bounded by $pr_i$ or intersect
Figure 3.3: Example of order inconsistency of projecting 3D structure to different viewing directions. Line \( L_2 \) lies infront of \( L_1, L_3 \) and while the order from left to right in view 1 is \( L_1, L_3, L_2 \) for view 2 is \( L_2, L_1, L_3 \) with it. As the projective transformation is injective and ordering is preserved, in the image plane the 2D point would have then been inside the triangle or the 2D line segment would have been bounded or intersected by the triangle respectively. This is however not possible according to the definition of the close constrained triangulation construction. QED

Figure 3.2 illustrates that the constrained triangulation in a single view results in a 3-space mesh, \( M_j \), which is consistent with the feature visibility. This is guaranteed to hold provided the projection, \( P_j \), is injective which is the case for perspective projection in conventional cameras.

### 3.4 N-view Reconstruction

In general for multiple views of a 3D scene there is no single 2D plane to which the scene features can be injectively projected without reordering of the features \( F \) with respect to one or more views (figure 3.3). As these assumptions made for a single view no longer hold, no consistent 3D model can be built from planar triangulation and backprojection.

For two views \( i \) and \( j \) a consistent model can be reconstructed by integrating the single
consistent models $M_i$ and $M_j$ in such a way that all triangles in the resulted model satisfy the visibility constraints of both views $C_{ij} = \{C_i \cup C_j\}$. This approach can be extended to reconstruct a consistent model for N-views, $M$, by progressively integrating new local models $M_i$ into the existing consistent global model $M$. The final model $M$ will then satisfy the union of all visibility constraints for the N-views, $C = \{C_i\}_{i=0}^{N-1}$.

The general N-view algorithm can be stated as follows:

1. Initialise the global model, features and visibility constraints as empty sets: $M = \{\emptyset\}$, $F = \{\emptyset\}$ and $C = \{\emptyset\}$.
2. Update features $F$ based on the corresponding estimates in frame $i$.
3. Build a consistent local model for the $i^{th}$ view, $M_i$, by constrained triangulation of the visible features, $F_i$, in a plane orthogonal to the view direction such that $M_i$ is consistent with $C_i$.
4. Remove triangles in the global model $M$ which violate the viewpoint constraints for the $i^{th}$ view $C_i$ so that the resulting model $M'$ is consistent with the combined constraints $C' = \{C \cup C_i\}$.
5. Remove triangles in $M_i$ which violate the viewpoint constraints of the global model $C$ so that the resulting model $M'_i$ is consistent with the combined constraints $C'$.
6. Add triangles from the consistent local model $M'_i$ into the consistent global model $M'$ to form a new global model $M = \{M' \cup M'_i\}$ which satisfies the new global constraints $C = C'$.
7. Repeat steps 2-6 for all N views to integrate the new local models, $\{M_i\}_{i=1}^{N-1}$, for each view to produce a global model $M$ which is consistent with the union of feature visibility constraints from all views $C = \{C_i\}_{i=0}^{N-1}$.

Although the algorithm is based on the incremental addition of new views the visibility constraints for all views are applied symmetrically to each triangle in the model and so it can be proved that the general N-view algorithm for reconstructing a consistent 3D scene representation is order independent (Appendix A). It should be noted that the visibility constraints are cumulative such that new constraints are only added and not removed from the global set $C$. 

3.5 Proof of convergence

Given a model $M$ which is consistent for the set of visible features $F$ over $N$ views it can be proved that as additional views are incorporated the model converges towards an approximation of the true scene surface $S$.

**Proposition 2:** If a model $M$ is consistent over $N$ views then as $N \rightarrow \infty$ the reconstructed model $M$ approximates a subset of the real scene surfaces $M_{\infty} \subseteq S$.

**Proof** A consistent model $M$ consists of two categories of triangles: (a) Real triangles which are planar approximations of real surfaces $M^r = \{t^r_i\}_{i=0}^{N_{tr}}$ where $M^r \subseteq S$; and (b) Virtual triangles which occur at occlusion boundaries $M^v = \{t^v_i\}_{i=0}^{N_{tv}}$. The union of real and virtual triangles is the model $M = \{M^r \cup M^v\}$. By definition the set of real triangles $M^r$ correspond to (opaque) surfaces in the real scene $S$ and therefore can not be intersected by a visibility constraint from any viewpoint. Therefore, this model is consistent for all possible viewpoints. Given a consistent model $M$ for $N$ views, applying the visibility constraints from a new view $C_{N+1}$ will only result in elimination of virtual triangles from the set $M^v$. Virtual triangles will be eliminated provided one or more scene features is visible on the other side of the triangle. A virtual triangle for which no scene feature is visible on the other side from any view is a valid planar approximation of the real surface and therefore belongs to the set of real surfaces $M^r$. As the number of views increases all virtual triangles will be eliminated $M^v_{\infty} \rightarrow \{\emptyset\}$. The resulting model $M_{\infty}$ will therefore converge to the set of triangles which correspond to real scene surfaces $M_{\infty} \rightarrow M^r \subseteq S$. QED

3.6 N-View Reconstruction Algorithm Complexity

The N-View reconstruction algorithm as formulated is an incremental algorithm where a new model $M_i$ is computed independently for each view and integrated with the global model, $M$. The analysis of the total complexity of this methodology is thus better
3.6. N-View Reconstruction Algorithm Complexity

determined by examining the computational requirements for each of the algorithm stages over integration of a single new view.

Starting with step 3, the process involves construction of the local model $M_i$ for the set of visible features $F_i$ in view $i$. This operation requires projection of $F_i$ to the image plane, constrained 2D triangulation over the projected primitives and backprojection to 3D space. Assuming an average number of features $N_{Fav}$ observed in each view then each of the projections has a $O(N_{Fav})$ complexity while the triangulation procedure can be computed with $O(N_{Fav} \log N_{Fav})$ complexity.

Step 4 ensures consistency of the global model $M$ with respect to the visibility constraints $C_i$ associated with features $F_i$ visible in view $i$. The number of these constraints is therefore proportional to $N_{Fav}$. As the process involves intersection tests between each triangle in the model and every visibility constraint the computational costs of directly applying the test will thus be $O(N_M N_{Fav})$ where $N_M$ is the number of triangles in $M$. Using a spatial partitioning scheme this number can be reduced to $O(N_{Fav})$ making the overall complexity of the step $O(N_{Fav}^2)$.

The use of a constrained triangulation to build each local model ensures that the mesh vertices ($N_v$ the number of vertices) coincide with the points and segment endpoints of the projected $F_i$. Furthermore, it is known [70] that each polygon with $n_b$ boundary vertices and $n_i$ internal ones can be triangulated with $n_b + 2n_i - 2$ triangles. As $n_b + n_i = N_v$ it is safe to assume that the number of triangles in an average local model is $O(N_{Fav})$. Step 5 of the reconstruction theory checks consistency of the local models $M_i$ over the set of global visibility constraints $C = \{C_j\}_{j=0}^{i-1}$ with complexity $O(N_{Fav} N_C)$ where $N_C$ is the number of accumulated constraints in $C$. Considering that $C_i$ the visibility constraints in a single frame are proportional to $N_{Fav}$ then over $N$ views $N_C = O(N N_{Fav})$ and therefore the overall step complexity can be stated as $O(N N_{Fav}^2)$.

Finally integration of the consistent local model $M'_i$, and global model, $M'$, is required. To ensure order independence the integrated model, $M$, must be the union of $M'_i, M'$ and so redundant triangles have to be identified and deleted. This process involves $O(N_{M'} N_{M'})$ triangle-triangle comparisons where $N_{M'}$ and $N_{M'}$ are the number of
triangles in the global and local models respectively. Assuming that $N_{M'_i}$ is $O(N_{F_{av}})$ as $M'_i \subset M_i$ and correspondingly $N_{M'}$ is $O(N_{F_{av}})$ the complexity of integration is $O(N_{F_{av}}^2)$.

It has been shown that the complexity of the algorithm for a single ‘average’ view is $O(NN_{F_{av}}^2)$. Based on the fact that this process is repeated $N$ times the overall computational cost for the entire sequence will be $O(N^2N_{F_{av}}^2)$. This cost is prohibitively expensive due to the dependence on the square of the number of views involved. Practically this means that as the number of views increases the cost per new view increases linearly with the number of already processed views. To indicate this, in a sequence of one hundred images with an average of fifty features per view integrating the 100th view will require approximately 250000 visibility tests. Therefore, practical reconstruction for long image sequences requires techniques for which the computational cost is not dependent on the number of previous views.
Chapter 4

Incremental Reconstruction Algorithm

The geometric sparse data reconstruction theory presented in chapter 3 addressed the problem of building surface models from sparse feature estimates together with their visibility across N-views. To achieve efficiency over extended sequences the system must satisfy two criteria:

1. It should only rely on a sparse set of features as input.

2. Integration of a single frame \( I_t \) should be independent of any information available in past frames \( I_{(s)}^{t-1} \).

This chapter describes an algorithm based on the general theory presented that successfully achieves both of these goals. The algorithm efficiently processes data as they are acquired from each new view and builds an updated 3D model that provably converges to the real world environment.

4.1 Reconstruction theory simplification

Reconstructing a 3D representation of a scene is a process that involves capturing and processing a large number of images during exploration of the whole real site.
However, the modelling theory proposed in the earlier chapter exhibits for each frame a complexity dependent on the number of previously processed views. To overcome this problem an efficient algorithm has been developed that progressively integrates new frames while maintaining a global model at each time, independent of the length of the sequence.

The point that causes the bottleneck in the process of a frame $i$, in the presented theory, is the step where verification of the model is required based on the accumulated visibility constraints $C = \{C_j\}_{j=0}^{i-1}$ from all previous frames. Circumventing this stage by only checking the models validity against the visibility constraints $C_i$ is the key idea towards an algorithm with efficient computational and storage cost. This simplification has also led to changes in the integration process in order to ensure convergence of the algorithm to the real world structure. Thus instead of incorporating all non redundant local model triangles only triangles associated to features $F_{i}^{new} \subseteq F_i$ that become visible for the first time in frame $i$ have to be integrated.

Under these simplifications the complexity of both the visibility check and the model integration has been reduced as the cost is not dependent on the number of previous views. However, potentially more views are required for the model to converge to the real scene topology due to hypothesis of new triangles that are locally consistent but may violate visibility constraints from earlier viewpoints. Furthermore, the order independence characteristic of the reconstruction theory is no longer guaranted. Nevertheless, considering $F_{i}^{ex} \subseteq F_i$ the subset of the features in a frame $i$ that have already been seen in a previous frame the following proposition still holds:

**Proposition 3:** For a closed-scene with features $F$ the global model $M$ reconstructed by only applying the visibility $C_i$ for visible features $F_i$ in each new view $i$ converges to a subset of the real scene surfaces as the number of views increases $M_\infty \subseteq S$

**Proof:** If the system is closed then as the number of views increases the set of new features $F_{i}^{new}$ visible in each frame converges to zero, $F_{i}^{new} \notin F = \{\emptyset\}$. Consequently the set of new triangles integrated to the global model converges to zero. Each new view
results in a new set of visibility constraints, $C_i$, for features previously incorporated into the model $F_i^{\text{ex}} \cup F_i^{\text{new}} = F_i$. Following the proof of Proposition 2, the $i^{th}$ view visibility constraints, $C_i$, will eliminate only virtual triangles $M^v \in M$. As the number of views increase the global model will converge to the set of triangles approximating real surfaces $M_\infty = M^r \subseteq S$. QED

\section{4.2 Incremental Algorithm}

A modified reconstruction algorithm which only applies the visibility constraints $C_i$ for features $F_i$ at the $i$ view has been developed. This algorithm can be summarised in the following steps:

1. Initialise the global model and features to empty set : $M = \{\emptyset\}$, $F = \{\emptyset\}$.

2. Build an initial model $M_0$ using constrained triangulation on the set of features $F_0$ reconstructed from the first view.

3. For each new view $i$ obtain a new set of features $F_i$ and the corresponding visibility constraints $C_i$.

4. Update the 3D position of features $F_i^{\text{ex}}$ which already exist in the global model $M$ for which a new estimate has been computed.

5. If the set of features $F_i^{\text{new}} \subseteq F_i$ that appear for first time in view $i$ is not empty

   (a) Eliminate triangles in $M$ that violate the visibility constraints $C_i^{\text{new}}$ imposed by features in $F_i^{\text{new}}$ and obtain $M'$.

   (b) Build a consistent model for the $i^{th}$ view $M_i$, by constrained triangulation of the visible features $F_i$, in a plane orthogonal to the view direction.

   (c) Integrate non-redundant triangles from $M_i$ (associated to $F_i^{\text{new}}$) to $M'$.

6. Eliminate triangles in $M'$ that violate the visibility constraints $C_i^{\text{ex}}$ associated with features $F_i^{\text{ex}}$ which have already been seen in a previous view and obtain the global model $M$. 
Figure 4.1: Illustration of reconstruction for two consecutive views. (a),(b),(c) Step 2. (d),(e) Step 4. (f),(g) Step 5a.
4.2. Incremental Algorithm

Figure 4.2: Illustration of reconstruction for two consecutive views. (a),(b) Step 5b. (c),(d) Step 5c. (f),(g) Step 6.
A graphical representation of the algorithm for two consecutive frames of a synthetic scene consisting of a box in the middle of a room is presented in figures 4.1 and 4.2. Initially an image $I_0$ is captured and the estimated features $F_0$ superimposed are illustrated in 4.1(a) (in practice at least two frames are required for estimating any real world 3D structure using an SFM algorithm). The constrained triangulation of the features in the viewing plane is computed 4.1(b) and backprojection to 3D space results to the local model $M$ presented flatshaded in figure 4.1(c).

The camera subsequently moves slightly to the left so that the side of the cube becomes visible. Figure 4.1(d) shows the existing measurements for features $F_1^{ex}$ already in the model imposed on this new view. Update on features $F_1^{ex}$ is subsequently performed, as it will be discussed in the following section (figure 4.1(e)).

As a number of features $F_1^{new} \not\in F$ (figure 4.1(f)) appear for the first time at frame $I_1$, validation of the model against visibility of this feature set has to be tested. The resulting model $M'$ presented in figure 4.1(g) illustrates the elimination of the triangles connecting the left front side of the box with the back wall.

Hypothesising new structure is performed by a 2D triangulation of all features $F_i$ visible in frame $I_1$ (figure 4.2(a)). The local model $M_1$ obtained from this process is presented in figure 4.2(b). The next step in the algorithm involves integration of this local model $M_1$ with the existing global model $M'$. Two different views of the resulting model are shown in figures 4.2(c) and 4.2(d). Finally, the visibility is checked against existing features $F_1^{ex}$ yielding the final model $M$ which is depicted in figures 4.2(e) and 4.2(f). This demonstrates the elimination of triangles that connect the box to the back wall.

This process applied to a number of images captured from different viewpoints around the room progressively builds a 3D model that converges to an accurate and consistent approximation of the scenes geometry and topology.

### 4.3 Algorithm implementation

A system aiming in the autonomous navigation of a mobile robot platform and the incremental modelling of its surrounding environment has been presented in section
1.1. To achieve reconstruction of scene geometry a feature based structure from motion approach is used [60, 62]. This method relies on detecting and tracking 2D image features over a sequence of frames and reconstructing feature positions and camera pose under the rigidity assumption. In general, different types of features are used for different contexts but common choices include corners and lines.

A higher level feature than corners and lines is junction. Junctions combine the characteristics of both these features as they consist of a set of line segments and the point associated with the corner structure. Junctions are more robust for matching, outline the boundaries of real world objects especially in man made environments that are mostly built with planar faces and provide clues for feature groupings [84, 85]. These are properties applicable to the described system and thus junctions are the underlying structure primitives that has been decided to be utilised.

Since junctions are geometrically complex image primitives their principal drawback is their absence in images of uniform scene regions. Nevertheless, deriving meaningful 3D measurements using passive sensing on such areas is difficult anyway because of the lack of features for correspondence between views. The type of junctions that are used in this study are V-junctions, Y-junctions and T-junctions. The input to our surface modelling system [57, 55, 56] is thus the estimates of the 3D lines and points associated to these junctions along with estimates on the camera motion parameters and all the corresponding covariances associated.

Further implementation details for each of the algorithms steps except the constraint triangulation that appears separately in the next chapter are presented in the reminder of this section.

### 4.3.1 Feature Update

For every new frame \(i\) processed, a set of 3D features \(F_i\) is estimated from SFM. A subset of these features \(F_{i}^{\text{ex}} \subseteq F_i\) have already been seen from a past viewpoint. Consequently they have previously been incorporated into the reconstructed model \(M\). The remaining features \(F_{i}^{\text{new}}\) become visible for first time at frame \(i\). The two subsets are complementary and thus \(F_i = F_{i}^{\text{ex}} \cup F_{i}^{\text{new}}\). Due to tracking the same feature across
Chapter 4. Incremental Reconstruction Algorithm

multiple frames its SFM estimate improves and its variance decreases [61]. In addition the visible segment of the line changes and hence line endpoints must be updated. Thus a continuous model update is necessary for features $F_i^{ex}$ in each frame $i$. There are two main steps in this update process:

1. Check for vertices that have to be linked into a junction and update the model accordingly.

2. Update feature $F_i^{ex}$ based on the new corresponding 3D estimates.

In a typical frame not all junctions connecting features visible in the image can be extracted. This can be caused from a possible failure of the low level feature extraction process or occlusions resulting from the scene topology. In either case if any such junction (V-junction, Y-junction) appears in a later frame it must be integrated into the scene model. A problem like the one presented in figure 4.3 may arise. Such situations are resolved in step 1 of the update process.

In the scene example of figure 4.3(a) from the initial viewpoint $i$ only T-junctions are identified between lines $a, b, c$ and the surface $S$, while the true V-junction remains hidden behind $S$. This will yield a local model like the one presented in figure 4.3(b) where the surface $S$ is connected with the lines $a, b, c$ at points $A, B, C$ respectively. Assume then that the camera moves to viewpoint $i + 1$. The endpoints $A, B, C$ of the three associated features $a, b, c$ now appear to represent the same scene structure $P$ and so have to be integrated to a single point. This is a process that involves all reconstructed structure associated with any of the three vertices $A, B, C$ being consistently reordered and linked to a single new point $P$. Cases like this occur very often especially when the scene exhibits significant occlusions.

In the second step the line features $F_i^{ex}$ are updated based on their new measurements computed from SFM. Endpoints of lines participating in a junction are updated to the corresponding new junction positions. Endpoints not associated with a junction are first projected onto the corresponding new line estimate. The feature is then updated to the greatest segment that can be formed from the endpoints of the new and the projected endpoints of the existing estimate.
4.3. Algorithm implementation

Figure 4.3: (a) Scene from viewpoint $i$. Only T junctions are extracted. (b) Reconstructed model based on the available junction information. (c) New viewpoint reveals a Y-junction between already reconstructed structure.

4.3.2 Feature visibility

A hypothesise and verify strategy is the basis for the proposed reconstruction approach. Surfaces are hypothesised based on locality criteria using a constrained triangulation process. However, the limited visibility from a single viewpoint $i$ is arbitrary and therefore results in triangles that do not correspond to real scene surfaces. These triangles satisfy the visibility constraints $C_i$ associated with features $F_i$ of the particular viewpoint. Thus the methodology for invalidating such triangles rely on checking visibility from different views. Assuming $F_i$ is the set of features against which the model has to be tested, the process can summarised as:

1. For each feature $f_j \in F_i$
   
   (a) Identify triangles in the model that when projected to the $i^{th}$ image plane overlap with $f_j$.

   (b) Check the visibility constraint $C_j$ of feature $f_j$ for each of the associated identified triangles relative the $i^{th}$ viewpoint.

In step 1a, potential inconsistent triangles are identified as the ones that overlap with the projected feature $f_j$ in the image plane. Each such triangle has two 2D point
entries assigned that correspond either to the intersection point between the projected line feature $f_j$ with one of the triangle edges or the endpoint of the feature bounded by the triangle.

In the second step, each triangle from the set of candidate invalid ones associated with a feature $f_j \in F_i$ has to be checked for visibility. In the 2D image plane each of the two point entries related to the triangle also lies on the projected feature $f_j$. Knowing the 3D position of both the triangle and the $f_j$ allows the estimation of their relative depth in 3-space with regard to the camera centre. Assuming intrinsic camera parameters $f_x, f_y, x_0, y_0, P_i = (x, y)$ the 2D point entry on the image plane, and $A, B, C, D$ the parameters of the plane defined by the 3D triangle $t$, then the point $P_t = (x_c, y_c, z_c)$ that corresponds to the intersection of the ray passing through the camera centre and the point $P_i$ with the triangle $t$ can be computed as:

$$x_c = \frac{(x - x_0)z_c}{f_x} \quad (4.1)$$
$$y_c = \frac{(y - y_0)z_c}{f_y} \quad (4.2)$$
$$z_c = \frac{-Df_y}{Af_y(x - x_0) + Bf_x(y - y_0) + Cf_xf_y} \quad (4.3)$$

Depth of features can then be classified relative to triangles into one of the following categories:

1. Feature behind the triangle if both line entries lie behind the corresponding entries for the triangle.

2. In front of the triangle if both lines 3D entries are in front.

3. Coincides with the triangle if all four 3D entries have similar depth values.

4. Intersects the triangle if one of the lines 3D entries lies in front and one behind the triangle.

Triangles classified in categories (1) and (4) violate the visibility constraint formed by the examined feature and so have to be eliminated from the reconstructed model.
4.3.3 Model Integration

The incremental reconstruction of the scene model is a principal characteristic of this algorithm. As the camera moves around the environment previously estimated surface hypothesis are invalidated according to the new feature visibility and new structure which appears for the first time is integrated into the model. At every viewpoint \( i \) where new features are extracted (\( F_{i}^{\text{new}} \neq \emptyset \)) a local 3D model is built using a constrained triangulation process. As already proved in section 3.3 this model is locally consistent in the specific view. Therefore the set of triangles that span the space between features \( F_{i}^{\text{new}} \) and features that already exist in the global model \( M \) are also locally consistent as no visibility constraint \( C_{i} \) is violated.

In the model integration process all triangles in the local model \( M_{i} \) associated to each of the features in \( F_{i}^{\text{new}} \) are initial identified. A triangle is considered to be associated to a feature if at least one of its vertices coincide with a feature endpoint. In case the triangle is not already part of the \( M \) (associated to more than two new features) then it is integrated into the model.

For every new V-junction and Y-junction identified, triangles are also hypothesised to connect each pair of participating line features. These junctions in practice outline the boundaries of real world objects and their integration results in triangles that give a correct surface approximation. Nevertheless the validity of these triangles is tested against the visibility of features in \( F_{i}^{\text{ex}} \) in the last step (6) of the algorithm.

4.4 Recursive N-View Algorithm Complexity

Examination of the recursive algorithm complexity will be presented in terms of computational requirements for the processing of each single frame \( i \). To facilitate the description and be consistent with the complexity analysis of the geometric reconstruction theory we define \( N_{F} \) the number of features in the model and \( N_{F_{\text{av}}} \) the average number of features in each frame.

The first step of the algorithm involves updating existing features \( F_{i}^{\text{ex}} \) based on their new estimates and the junction information. Both potential linking of vertices and end-
point updates are constant time operations for each of their corresponding features. As the number of features is proportional to $N_{F_{av}}$ the whole process exhibits an $O(N_{F_{av}})$ complexity.

Following the update and in case any new structure $F_{i}^{new}$ has become visible for the first time in frame $i$ several steps are performed. The visibility consistency of the model $M$ is tested relative to $F_{i}^{new}$; the local model is built and integration of local and global model is applied. The complexity of testing visibility will be studied irrespective of the set of features imposing the constraints while constructing the local model involves a constrained 2D triangulation process with $O(N_{F_{av}}^2)$ worse case complexity that will extensively examined in the next chapter.

Assuming that $F_{i}$ is the set of features placing the visibility constraints against which the model has to be checked, its cardinality can be considered in the order of $O(N_{F_{av}})$. Presuming that the number of triangles in the model is proportional to $N_{F}$ [70] identifying potential invalid polygons requires one to one 2D intersection tests between the projected features and the triangles and so is an $O(N_{F_{av}}N_{F})$ process. Finally, considering that the number of these possible inconsistent triangles is only a fraction of the whole set but still in the order of $O(N_{F})$ the process of testing all relative depths of each triangle against each feature in $F_{i}$ will have $O(N_{F_{av}}N_{F})$ complexity.

The overall complexity of checking the model’s consistency in both steps 5a and 6 of the recursive algorithm has been estimated as $O(N_{F_{av}}N_{F})$. The dependence of the process on $N_{F}$ however indicates that the scene complexity affects the algorithm performance.

In order to circumvent this problem a spatial partitioning has been developed. From each viewpoint the pyramid formed from the camera centre and the corners of the image plane projected in 3-space has been computed and only the part of the model that is inside this pyramid is considered. This pyramid effectively denotes the visible region for the camera or view frustum so that any polygonised structure outside this cannot be affected from local feature visibility constraints. This scheme can be used to reduce the number of features in the global model $N_{F}$ to an order proportional to $N_{F_{av}}$ and thus the complexity of the model verification using visibility becomes $O(N_{F_{av}}^2)$. The final integration step involves incorporation of the triangles associated to new
structure from the local model $M_i$ to $M'$. Triangles associated to features $F_i^{new}$ cannot already exist in the global model and thus search for redundancy is limited between them as a triangle may be associated to more that one endpoint of a feature in $F_i^{new}$. Assuming that the number of such triangles is proportional to $N_{F_{av}}$ the process has an $O(N_{F_{av}}^2)$ complexity. Redundancy for hypothesised triangles connecting pairs of junction lines involves searching among the $N_F$ global triangles in the current field of view and so also has an $O(N_{F_{av}}^2)$ complexity.

Processing a single frame using the incremental reconstruction algorithm developed has shown to exhibit an $O(N_{F_{av}}^2)$ complexity. Assuming that an entire sequence of $N$ views have to be processed the overall reconstruction complexity becomes $O(NN_{F_{av}}^2)$. This reveals a clear improvement over the theoretic modelling algorithm introduced in the previous chapter, as a linear complexity over the number of frames is maintained ensuring efficient applicability to extended video sequences.
Chapter 5

Surface estimation using 2D constrained Delaunay triangulation

This chapter presents an approach for building a triangular mesh representation of the real world scene as it has been captured from a single viewpoint. The method starts with the 3D features reconstructed from the SFM process and makes use of the planar Delaunay triangulation to produce a polygonisation that includes the projections of these features to the image plane as part of the model. The computed 2D mesh is subsequently backprojected to 3-space resulting in a triangular piecewise continuous surface model in which original features are preserved. An important feature of this surface hypothesis methodology is the low computational cost as all the processing is performed in 2D.

The choice of the Delaunay structure is initially justified and an overview of different approaches proposed for its construction are presented. A fast and simple algorithm for the incremental constrained Delaunay triangulation of a set of point and line segments has been developed and described along with its complexity analysis.
5.1 Delaunay Triangulations

Mesh generation on a 2D space has been examined extensively over the last fifty years due to the wide area of potential application such as geology, cartography and more recently finite elements and computer graphics. Given $N$ points on a plane an upper bound of $10^6N$ different triangulations exist [34] making the straightforward idea of exhaustive search for the best scheme prohibitively expensive. Therefore, interest has focused on triangulations that optimise some measure of quality. Quality criteria commonly imposed are related to the shape of the triangles, the edge lengths and the internal angles.

A unique triangulation that optimises several such criteria simultaneously is the Delaunay triangulation. The planar Delaunay triangulation of a set $S$ is most easily introduced by reference to the Voronoi diagram of $S$. The Voronoi diagram is defined as the tessellation of the plane to convex polygonal cells each one corresponding to an input point $s \in S$ and enclosing the region where all points are closer to $s$ than every other point in $S$. The Delaunay triangulation is a geometric dual of the Voronoi diagram. On each vertex of the Voronoi diagram three territorial boundaries meet. These three regions around the vertex belong to the three points that form a Delaunay triangle. An example of a Voronoi diagram and the corresponding Delaunay triangulation is illustrated in figure 5.1. A detail description of properties like max-min angle, min-max circumscribed circle, min-max enclosing circle all satisfied by the Delaunay triangulation as well as relations with the Voronoi diagram has been presented by Fortune [24]. Due to these properties Delaunay triangulations are widely used in engineering analysis with finite element techniques where uniform triangles are required and computer graphics to optimise rendering.

The hypothesis generation step of the reconstruction algorithm presented in chapter 4 requires for each new frame $i$, the projection of the estimated 3D features $F_i$ to the plane orthogonal to the view direction, planar triangulation and backprojection in 3-space. As perspective projection $P()$ is injective the order between features $P(F_i)$ in the plane is preserved relative to the corresponding ordering in the scene. Hence it has been proved (Proposition 1) that this methodology results in a consistent model
5.1. Delaunay Triangulations

Figure 5.1: The Voronoi and Delaunay structures of a set of points. Delaunay triangles are sketched with bold lines.

$M_i$ according to Definition 4. How close this model fits to the real scene however, is principally dependant on the 2D meshing process. The criterion which has been chosen for connecting features is locality. Although this is a heuristic it is reasonable to assume that features close to each other are most likely to belong to the same surface.

Based on this optimisation measure the Delaunay triangulation is by definition the required structure as each Voronoi vertex is the circumcentre of its corresponding Delaunay triangle and no other point may lie within this circle. The empty circumscribed circle property can be used both as a construction medium [49] for the triangulation and as definition

**Definition 5 (Delaunay Triangulation):** A triangulation $T$ of a set of planar points $S$ is a Delaunay triangulation if for each triangle $t_i \in T$ the circumscribed corresponding circle does not contain any other vertex of $S$.

This definition considers a planar set of features $S$ that consists exclusively of 2D points. In order to build a single view model which contains all the estimated real scene primitives the projected points and lines features have to be preserved in the triangulation. Unfortunately, creating a mesh that includes point and line constraints
Chapter 5. Surface estimation using 2D constrained Delaunay triangulation

Figure 5.2: Constrained triangulation of set of points B,D,E and line segments AC.
(a) As the circle through A,B,C bounds D and conversely the circle through A,D,C encompass B no triangulation of the input set can be achieved that satisfy the empty Delaunay circle criterion for the feature set. (b) Conforming Constrained Delaunay Triangulation by introducing new points F,G. (c) Constrained Delaunay Triangulation.

and simultaneously conforms with Definition 5 is not always possible as illustrated in figure 5.2(a).

The problem of incorporating line segments as constraints in the triangulation has been addressed by the conforming and constrained Delaunay triangulations. Assuming $V$ to be the set of points and $E$ the set of edges belonging to $S$ then a conforming Delaunay triangulation is a triangulation over an augmented set of points $A \supseteq V$ where $A \cap V$ points lie on $E$ and the Delaunay property holds for each triangle. Intuitively the idea is to split the original constraint edges into shorter segments by introducing new points in such a way that the resulted mesh is Delaunay 5.2(b). Several algorithms have been developed [32, 26, 30, 15, 66, 35, 77] to construct such a triangulation. Although the resulting mesh is well behaved, as all the Delaunay properties stand, the number of triangles is significantly increased. In the case of triangulating real scene features this process would also introduce new features which do not correspond to any real scene structure.
The alternative approach is the Constrained Delaunay Triangulation (CDT) which is the best approximation of the Delaunay triangulation (figure 5.2(c)) applied on the original set of features \( S = (V, E) \). Assuming visibility between two points \( v_i, v_j \in V \) when the segment \( \{v_i, v_j\} \) does not cross any edge in \( E \) then the CDT can be defined [25] as:

**Definition 6 (Constrained Delaunay Triangulation):** The Constrained Delaunay Triangulation contains the edge \( \{v_i, v_j\} \) between two input vertices, if and only if \( v_i \) is visible to \( v_j \) and some circle through \( v_i, v_j \) contains no other input point visible to segment \( \{v_i, v_j\} \) in its interior.

Existing algorithms for building the CDT can be classified to batch [50, 10, 16, 17] and incremental [11, 88, 6, 2, 54] with the difference lying on the availability of the feature set during the triangulation process. Thus incremental methods are based on stepwise addition of features while maintaining a Delaunay triangulation and batch rely on the existence of the whole set from the beginning. As the CDT generalises the definition of Delaunay triangulation in order to impose certain constraints in the mesh no additional triangles are generated and the Delaunay property holds locally for quadrilaterals formed from adjacent triangles that do not share a constrained edge.

### 5.2 Constrained Delaunay Triangulation

A new incremental planar Constrained Delaunay Triangulation algorithm has been developed in order to hypothesise the surfaces that span the space between the computed scene’s cloud of features. The algorithm incorporates new primitives as constraints applied to the existing mesh while maintaining a Delaunay triangulation at each time. The description has been separated to the process of adding points and line segments.

#### 5.2.1 Point Insertion

The process of incorporating points to the mesh is similar to an unconstrained incremental Delaunay triangulation [49]. Initially three arbitrary points are assumed such
that all input features lie inside the triangle they form. For each new point $p$ the insertion algorithm into the existing triangulation $T$ can be summarised as:

1. Locate the triangle $t \in T$ enclosing $p$.

2. Split $t$ to the three subtriangles formed by connecting $p$ with its vertices and delete it.

3. For each of the three new triangles put the edge opposite to $p$ in the stack $S$ of candidate invalid Delaunay edges.

4. While $S$ is not empty
   
   (a) Remove the first edge from the stack and check the Delaunay property in the quadrilateral formed by its two adjacent triangles.
   
   (b) If $p$ is inside the circumscribed circle of its opposite triangle in the quadrilateral then swap the diagonal and place the edges not adjacent to $p$ in the two newly created triangles in the stack $S$.

The resulting triangulation $T'$ is Delaunay and contains the union of the original and new points. This is a standard process commonly adopted by most of the incremental Delaunay triangulation algorithms constrained or not. In fact not only the Delaunay but other optimal 2D triangulations over different criteria can be achieved using this edge swapping methodology [95]. An example of adding a point to an existing mesh is presented in more detail in appendix B.1.

5.2.2 Line segment insertion

Imposing line segment constraints on the mesh is of great importance in building a realistic scene representation as features that outline significant surface boundaries in 3-space are actually preserved in the resulted estimated model. Adding a new line constraint $C$ is closely related to the point insertion as the developed algorithm can be summarised in two steps, incorporate line endpoints and impose the actual line segment.
5.2. Constrained Delaunay Triangulation

The first of these steps is performed using the point insertion algorithm described in the previous section. The set of triangles intersected by the constrain $C$ are then identified. The edges bounding the region defined by these triangles form a polygon $P_C$ called the bounding polygon. All the diagonals of the bounding polygon are intersected by the constraint $C$. Using a recursive swap method these diagonals are flipped in such a way that they do not intersect $C$ and the bounding polygon is constrained Delaunay triangulated.

The algorithm to impose a constraint $C_i = \{p_1, p_2\}$ into an existing 2D Delaunay triangulation $M = \{V, E\}$ can be summarised as follows:

1. If constraint $C_i = \{p_1, p_2\}$ is not an edge $C_i \notin E$ of the mesh $M = \{V, E\}$ then insert the endpoints $p_1, p_2$ into $M$, using the algorithm presented in section 5.2.1 such that $V' = V \cup \{p_1, p_2\}$ and $E' = E \cup C_i$.

2. Create a list $L$, of intersections between edges $\{e_j\} \in E$ and the constraint $C_i$ sorted according to their order of intersection along $p_1$ to $p_2$.

3. Initialise an empty stack $S = \{\emptyset\}$.

4. While $L$ is not empty take the first edge $e = (v_p, v_r)$.

   (a) Let the quadrilateral $q_e = (v_p, v_q, v_r, v_s)$ be formed by the two triangles adjacent to $e$.

   (b) If $q_e$ is convex call Recursive Delaunay Edge Swap process with $(S, e)$.

   (c) Else push $e$ into stack $S$.

The process Recursive Delaunay Edge Swap is recursive and it is called with $(S, e)$, a stack and a single edge. Its steps can be described as follows:

1. Swap edge $e = (v_p, v_r)$ to opposite edge $e' = (v_q, v_s)$ of quadrilateral $q_e$ formed by the adjacent triangles of $e$.

2. If $e'$ does not intersects $C$ then impose the Delaunay criterion to the polygon $B = (v_q, v_{q+1}, ..., v_{s-1}, v_s)$ formed by consecutive points $v_q$ along to $v_s$ on the bounding polygon $P_C$. 

3. while the stack is not empty $S \neq \emptyset$ and its top edge $e_{\text{next}}$ forms a convex quadrilateral then

(a) Pop $e_{\text{next}}$ from the stack $S$.

(b) Call Recursive Delaunay Edge Swap process with $(S, e_{\text{next}})$.

4. If $e'$ does intersects constraint $C$ then put $e'$ in stack $S$.

A simple example of imposing a single line segment on an existing constrained Delaunay mesh is presented in figure 5.3. Assume that the line constraint to be inserted is $AD$ and its bounding polygon after inserting the endpoints is illustrated in figure 5.3(a). In step (2) of the algorithm the list $L = \{BF, BE, CE\}$ of intersected model edges by the constraint is computed and in the subsequent step (3) the stack $S$ is initialised. As $L$ is not empty the first edge to be processed is $BF$. The quadrilateral $q = ABEF$ corresponding to $BF$ that is formed in step (4a) is convex and thus the Recursive Delaunay Edge Swap (RDES) process is called in step (4b). In the first step of this process edge $BF$ is swapped to $AE$ as illustrated in figure 5.3(b). Edge $AE$ does not intersect the constraint segment $AD$ but in step (2) nothing has to be further done due to polygon $B = AFE$ being a simple triangle. Stack $S$ is now empty and thus step (3) is not executed. Finally, $AE$ does not intersect $AD$ and so is not inserted to the stack.

The next edge in list $L$ is $BE$. However, the formed corresponding quadrilateral $ABCE$ is concave and so $BE$ is pushed in the stack $S$ in step 4(c) of the main process. The subsequent edge $CE$ in the list forms a convex quadrilateral $BCDE$ and so the RDES process is called in step (4b). Inside RDES $CE$ is initially swapped to $BD$ as it is shown in figure 5.3(c). The polygon $B = BCD$ is again a single triangle and does not have to be checked in step (2). In the next step (3) though the stack contains edge $BE$ and thus RDES is recursively called with the now empty stack $S$. Edge $BE$ is swapped to the original constraint segment $AD$ as illustrated in figure 5.3(c) and the process terminates as both polygons $AFED$ and $ABCD$ satisfy the Delaunay criterion.

A more detailed illustration of imposing a line segment over an example mesh is described in appendix B.2. This method is similar to [88] as it is based on sequential edge swaps. However, unlike that method the developed algorithm achieves a Constrained
Figure 5.3: Example of imposing a single line segment as a constraint on an existing mesh
Delaunay Triangulation over the bounding polygon as it iterates through the intersected edges. Furthermore, the use of a stack for the unprocessed edges significantly reduces the number of swapping tests and results in a single pass over the invalid diagonals of the bounding polygon.

5.2.3 Convergence of line segment insertion algorithm to CDT

The developed algorithm imposes a line segment over the mesh by a single pass through the diagonals of its bounding polygon. The principal advantage is that the resulting triangulation is the CDT of the subset of input features already incorporated to the mesh. To prove this two intermediate lemmas are used

Lemma 1 Each of the two subpolygons in which the constraint line segment $C$ splits the bounding polygon $P_c$ has at least one convex vertex.

Proof: Consider one of the two subpolygons $P_1$ and assume that it consists of $n$ sides. Then the sum of the internal angles of the polygon is $(n - 2)\pi$ [70] and thus no $n - 2$ concave internal angles may exist. As the constraint $C$ is one of the $n$ sides and by definition of $P_1$ the two angles associated to its endpoints are convex at least one of the remaining $n - 2$ vertices must be convex as well. The proof for the second subpolygon is similar QED.

Lemma 2 After processing the last of the intersected diagonals associated with one of the bounding polygons convex vertices $v_i$ the corresponding ear formed by this vertex $v_i$ and its two adjacent ones $v_{i-1}, v_{i+1}$ is removed.

Proof: First assume that the ear has a single diagonal. In such case the quadrilateral formed by the two triangles associated with this diagonal is convex and thus the swap of the diagonal results in the removal of the ear. Otherwise a set of diagonals $D$ connects the convex vertex $v_i$ with different vertices $V$ on the opposite side of the constraint $C$. Consider then the polygon $P$ formed from vertices $v_{i-1}, v_{i+1}$ and $V$ and
assume a labelling of $p_0 = v_{i-1}, p_1, \ldots, p_{n-1} = v_{i+1}$ where $p_1, p_{n-2}$ is the arrangement that correspond to the relative order in which their associated diagonal cut the ear of $v_i$. The diagonals $D$ are processed in this relative order as well.

A diagonal forms a convex quadrilateral if its endpoint in $P$ corresponds to a convex vertex of $P$. If this diagonal is swapped then the corresponding ear is removed from $P$ and the cardinality of $D$ reduced by one. In the situation where the endpoint of the diagonal in $V$ is a concave vertex of $P$ then it is inserted into a stack $S$. However, not all diagonals may belong to this category as using the same reasoning with Lemma 1 applied to polygon $P$ using segment $p_0, p_{n-1}$ as the base, there is always at least one convex vertex between $p_1$ and $p_{n-2}$. The removal of the ear related to this convex vertex $p_i$ by swapping the associated diagonal only affects the status (convex or concave) of the previous and the next vertex in $P$. Thus testing the diagonals in the order they appear in the stack $S$ insures that no convex vertex $p_j, j \leq i$ may exists. Ultimately, $P$ is reduced to a quadrilateral with only one diagonal left associated to $v_i$, a case examined initially. QED.

**Proposition 4** Insertion of a new line segment $C$ as a constraint in an existing Delaunay mesh $M = \{V, E\}$ of a set of point $V$ and line $E$ constraint features, based on the algorithm presented in the section 5.2.2 results in a new triangulation $T'$ that is the constrained Delaunay triangulation of $E \cup C$.

**Proof:** Line segment insertion starts by incorporating its endpoints to the mesh in case they do not already exist. This process is based on the algorithm by Lawson et at [49] which is proven to consistently add new vertices so that the empty circle criterion holds. Imposing the actual segment involves initial identification of the intersected edges and formation of the corresponding bounding polygon $P_C$. As the triangulation $M$ is Delaunay and no new vertices are added or removed the sides of this polygon $P_C$ are Delaunay edges and thus do not change in the final triangulation.

The algorithm proceeds based on incremental removal of ears from $P_C$. As there is always an ear at each side of the constraint $C$ and after processing the last of its associated intersected diagonals this ear is removed ultimately the bounding polygon
reduces to a convex quadrilateral whose diagonal swap results in the desired constraint edge. Consider the two subpolygons $P_1, P_2$ in which the $C$ divides $P_C$ and assume an ordering of their vertices $p_1^0, p_1^1...p_1^k$ and $p_2^0, p_2^1...p_2^l$ so that $p_1^0 \equiv p_2^0$ are the constraints first endpoint and $p_1^k \equiv p_2^l$ the second. Each ear $\{p_i^a, p_i^b, p_i^c\}, a \leq b \leq c, i = 1, 2$ that is removed splits its associated polygon $P_i$ into two subpolygons $P_i^1 = (p_i^0..p_i^a, p_i^b..p_i^c)$ and $P_i^2 = (p_i^a..p_i^c)$. Successive swaps of the diagonals of subpolygon $P_i^2$ then insures that it always remains Delaunay triangulated. The process continues until only a single triangle remain in each side of $C$ and thus with the swap of the last diagonal each of the two polygons $P_i^1$ degrade to the corresponding $P_i^2$ and thus are Delaunay triangulated QED.

5.3 CDT complexity analysis

Hypothesis of real scene surfaces are generated by 2D Constrained Delaunay Triangulation over the set of projected estimated point and line features on a plane orthogonal to the viewing direction. The processes of imposing these points and lines primitives to the mesh are closely related as the latter involves the former when the line endpoints are inserted.

The algorithm for adding a new point $p$ to an existing Delaunay mesh computationally can be separated to locating the triangle that bounds the point and the sequential edge swap for insuring the empty circle criterion. The method used for identifying the triangle which $p$ lies inside is the one proposed in Lawson [49] with complexity $O(N_t)$, where $N_t$ is the number of triangles in the mesh. However, in the average case the algorithm is as fast as $O(\sqrt{N_t})$ [93]. The number $N_t$ of triangles can be considered proportional to the number of features $n$ already in the mesh and so the algorithm is of the order of $O(n)$.

The algorithm starts with an initial triangle guess and the identification of its edge from which point $p$ is outside. As the triangles are CCW oriented this is the edge that has $p$ on its right. The search continues with the triangle adjacent to the previous on the extracted edge and until the point is on the left of all three edges of the new examined triangle. More sophisticated point location methods has been proposed [37] that may
be asymptotically faster but in the average case they do not perform much better than the adopted algorithm and their practical use is much more complex.

The recursive edge swap process complexity analysis is based on the observation that all new triangles in the updated Delaunay mesh are incident to the new point \( p \) and so testing the Delaunay criterion is a \( \Theta(k) \) where \( k \) is the number of edges incident to \( p \). However, in the worst case where the point \( p \) is connected to all other vertices this complexity becomes \( O(n) \). Considering that the point insertion algorithm is applied for the whole set of input features its overall complexity is \( O(n^2) \).

The line segment insertion algorithm can also be divided into the preprocessing and the update part. The first process involves adding the line endpoints which is performed similarly with the above point insertion algorithm and sorting of the mesh’s intersected edges in the direction from one endpoint towards the other. Assuming that the number of such edges is in the average case \( N_{fav} \) sorting can be executed in \( O(N_{fav}) \) time. This is achieved by choosing the triangle where the last point has found as the initial guess for the search of the subsequent point in the vertex locate step and then mimic the path from one endpoint to the other finding the intersected triangles.

The update part of the algorithm recursively achieves a CDT over the bounding polygon of the imposed constraint. The processes involved in the recursive step are check the convexity, swap the diagonal, check the Delaunay criterion on a quadrilateral and test edge intersections. An interesting feature is that all these processes are constant time and so the complexity of the update process is related to the number of times this recursive part is called. This number however is very difficult to compute exactly as it is dependant on the particular mesh structure. The algorithm certainly does a single pass through each of the intersected edges while edges that need further processing are inserted in a stack. The use of such structure insures that an edge is tested only if all edges previously above it in the stack have successfully swapped. Empirically it has been found that the average times of processing each edge is \( O(\log(N_{fav})) \) (in the example of appendix B.2 \( N_{fav} = 8 \) and the number of iterations is 15) and so imposing a line to the mesh is of the order of \( O(N_{fav}\log(N_{fav})) \).

Assuming that a total number of \( N \) lines have to be inserted in the existing mesh
the complexity of preprocessing step will be $O(N^2)$ while that of the update step $O(N N_{fa} \log(N_{fa}))$. Thus the bottleneck of the whole process is due to the point insertion algorithm while the line insertion part is asymptotically optimal.
Chapter 6

Robust Reconstruction

Any system aiming at reconstructing real world scenes should not only be based on a consistent geometric framework but should also be able to address the inherent problem of measurement noise. This chapter presents how uncertainty information on the estimated position of the 3D features provided by structure-from-motion can be utilised to achieve robust reconstruction in the presence of noise. In particular the focus is on the update and the visibility processes of the reconstruction algorithm introduced in chapter 4 where the use of uncertainty enables:

1. Propagation of feature uncertainty to incrementally update the model based on improved estimates of feature locations.

2. Robust visibility constraint test to avoid inconsistent elimination of valid triangles due to noise.

Initially the underlying geometric probability assumptions are described and an uncertainty representation for 3D points, lines and triangles is introduced. Based on this foundation the reconstruction algorithm presented in chapter 4 is extended to explicitly consider noise in the measurement estimates.
6.1 Geometric Uncertainty

An estimated geometric object can be considered as a random variable described by a vector \( \mathbf{x} \) which consists of the variables that we have chosen to parameterise it. Thus we can define its probability density function (pdf), \( p(\mathbf{x}) \), as the probability of the specific object \( \mathbf{x} \) in the corresponding parameter space. In this sense geometric uncertainty can be treated using classic probability theory [99].

In practice obtaining an explicit pdf is extremely difficult because of the complexity of modelling all the sources of errors. However, in many cases a reasonable assumption is that the pdf is Gaussian. This assumption can be justified if noise is caused by a large number of independent sources from the central limit theory. There is also a practical motivation for choosing the Gaussian distribution such that it can be fully specified by the first and second order statistics mean, variance which is the only information that needs to be propagated through the system.

This characteristic is very useful because the transformation of a pdf reduces to that of transforming its mean and covariance. In particular assuming a Gaussian random vector \( \mathbf{x} \) with mean \( \overline{\mathbf{x}} \) and covariance matrix \( P_x \) and a transformation equation \( \mathbf{y} = f(\mathbf{x}) \). In case \( f \) is a linear transformation of the form \( \mathbf{y} = A\mathbf{x} + b \) the mean and covariance of \( \mathbf{y} \) will then be:

\[
\overline{\mathbf{y}} = A\overline{\mathbf{x}} + b
\]
\[
P_y = AP_x A^T
\]  \hspace{1cm} (6.1)

Similarly [96], if \( f \) is not linear with Jacobian matrix \( J = \partial f / \partial \mathbf{x} \) relative to \( \mathbf{x} \), then the corresponding statistics up to the second order becomes [96]:

\[
\overline{\mathbf{y}} = f(\overline{\mathbf{x}})
\]
\[
P_y = JP_x J^T
\]  \hspace{1cm} (6.2)

A property of geometric uncertainty is that a physical representation can be given to random variables. For a 3D point \( \mathbf{x} \) its covariance can be visually described with an ellipsoid.
6.1. Geometric Uncertainty

Figure 6.1: Uncertainty representation for (a) Point $\mathbf{x}$ and (b) Line $\mathbf{l} = (\mathbf{x}_1, \mathbf{x}_2)$ and point $\mathbf{p}$ on the line.

This ellipsoid is centred at $\mathbf{x}$ and bounds the volume inside which $\mathbf{x}$ is expected to lay with a probability specified by $k$ as illustrated in figure 6.1(a).

Based on this 3D point covariance representation the line segment uncertainty can be derived as it can be defined by its endpoints $\mathbf{x}_1, \mathbf{x}_2$ combined to a vector $\mathbf{l} = (\mathbf{x}_1, \mathbf{x}_2)^T$. Obtaining the variance for any point $\mathbf{p}$ along the line segment is achieved by the use of a linear interpolation scheme. Point $\mathbf{p}$ will then be defined as

$$\mathbf{p} = \mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1)$$  \hspace{1cm} (6.4)

Equation 6.4 can be considered as a linear transformation of $\mathbf{l}$ in the form $\mathbf{p} = A(\mathbf{l})$ where

$$A = \begin{bmatrix} 1 - \lambda & 0 & 0 & \lambda & 0 & 0 \\ 0 & 1 - \lambda & 0 & 0 & \lambda & 0 \\ 0 & 0 & 1 - \lambda & 0 & 0 & \lambda \end{bmatrix} = ((1 - \lambda)I \ | \ \lambda I)$$  \hspace{1cm} (6.5)

According to equation (6.1) the mean $\overline{\mathbf{p}}$ and covariance $P_p$ will then be
Figure 6.2: Uncertainty representation for triangle $t = (x_1, x_2, x_3)$.

$$\mathbf{p} = A\mathbf{I} = \begin{bmatrix} (1 - \lambda)I & \lambda I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T = (1 - \lambda)x_1 + \lambda x_2 \quad \text{(6.6)}$$

$$P_p = APA^T = \begin{bmatrix} (1 - \lambda)I & \lambda I \\ 0 & \lambda I \end{bmatrix} \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} (1 - \lambda)I \\ \lambda I \end{bmatrix} = (1 - \lambda)^2 P_1 + \lambda^2 P_2 \quad \text{(6.7)}$$

where $P_1$ and $P_2$ are the covariance matrices of $x_1, x_2$, which are considered independent, respectively. As equation (6.7) is quadratic relative to $\lambda$ the uncertainty envelope around a 3D line can be visualised as an hyperbolic ellipsoid and is illustrated in figure 6.1(b). It should be noted that the minimum uncertainty along the line is between $x_1$ and $x_2$.

The SFM method adopted uses an internal minimal line representation with 4 degrees of freedom [59]. This results in a 3x3 covariance matrix (for each point) with the null space along the line direction. Thus the uncertainty over each endpoint can be represented with a 2D ellipse in the plane perpendicular to the line orientation.

Using a similar methodology with lines we can model the covariance for any point on a triangle surface patch. Using barycentric coordinates any point that lies inside the triangle can be linearly interpolated from the triangle vertices. Assume a triangle $t = (x_1, x_2, x_3)$ and a point $p_t$ on the triangle. The point $p_t$ can be expressed in barycentric coordinates as
6.2. Model update based on uncertainty information

\[ \mathbf{p}_t = u \mathbf{x}_1 + v \mathbf{x}_2 + (1 - u - v) \mathbf{x}_3 \] (6.8)

where \( u, v \in [0, 1] \). If each of the triangle vertices is treated as a 3D point with known mean and covariance then following the same reasoning with the line uncertainty representation, the mean \( \mathbf{p}_t \) and covariance \( P_{p_t} \) of point \( \mathbf{p}_t \) can be defined as:

\[
\begin{align*}
\mathbf{p}_t &= u \mathbf{x}_1 + v \mathbf{x}_2 + (1 - u - v) \mathbf{x}_3 \\
P_{p_t} &= u^2 P_{x_1} + v^2 P_{x_2} + (1 - u - v)^2 P_{x_3}
\end{align*}
\] (6.9)

A graphical representation of the uncertainty envelope around a triangle is illustrated in Figure 6.2. Again the minimum uncertainty is for points close to the barycentre of the three triangle vertices.

6.2 Model update based on uncertainty information

The incremental reconstruction system presented in chapter 4 updates the model according to new structure estimates as these become available for each new frame of the image sequence. The recursive SFM \[60\] not only provides measurements on the 3D positions of features but also computes their associated covariance which corresponds to the uncertainty related to these estimates under a specific confidence value. This input is used to impose junction information to the associated features and update feature positions and statistics.

6.2.1 Impose junctions

Junctions are first identified in the low level image feature extraction process and propagated through the SFM to the reconstruction process. As feature extraction is a local 2D process applied on the intensity image and SFM is a global optimisation procedure where junctions are not imposed as hard constraints, the computed 3D estimates of the junction associated features that are fed to reconstruction are not in exact agreement
with the corresponding junction position. Errors in the localisation of the junction in
the image or slight misplacement of the associated lines in 3D usually result in estimated 3D segments that either do not meet at all or intersect instead of sharing a common point.

Imposing the junctions on the endpoints of the corresponding participating lines is performed in 3D space. In case of a V-junction or Y-junction, lines are updated so that they meet at the point with the minimum square distance from all their corresponding endpoints. Assuming \( x_i, i = 1..n \) the endpoints of the lines participating to the junction and \( P_i \) their covariance matrices then the new junction point \( p \) and its variance \( P_p \) will be

\[
p = \frac{\sum_{i=1}^{n} x_i}{n} \quad P_p = \frac{\sum_{i=1}^{n} P_i}{n^2}
\]

The other type of junction that is considered is the T-junction. Assume the example of figure 6.3(b) where such a junction between two lines \( l_1 = (x_1, x_2), l_2 = (x_3, x_4) \) is illustrated. First the point \( P \) on line \( l_2 \) which is nearest to line \( l_1 \) is computed. In case \( P \) lies between the endpoints of \( l_2, x_4 \) is shifted to \( S \) which also lies on the line \( l_2 \) very close to \( P \) in the direction towards the other endpoint \( x_3 \). In this way splitting line \( l_1 \) during the constrained triangulation process as well as dealing with exact alignment of feature endpoints and the camera centre are avoided. The covariance of the new endpoint \( S \) is finally interpolated using equation 6.7.

Connecting noisy feature estimates by using the junction information significantly improves the quality of the reconstruction. In the cases where the line endpoints are close but do not meet a number of small and thin triangles would otherwise be hypothesised to close the gap between them. This is illustrated in the results of section 7.1.3 where reconstruction is performed without any junctions. In the opposite situation where the lines intersect in 2D, as presented in the example of figure 6.3(a), the visibility step of the reconstruction algorithm would have incorrectly removed triangles associated with line \( l_1 \). Furthermore, the principal assumption of the 2D triangulation process that its input set is a planar straight graph would have been violated.
6.2. Model update based on uncertainty information

Feature detection in real video sequences is highly dependent on the actual structure of the scene and the capture conditions (small baseline, illumination) and so it is not always realistic to assume that every possible significant feature is extracted. Thus planar intersections between the projected 3D feature estimates may still occur in cases where the corresponding junctions have not been estimated. Dealing with such problem involves addressing situations where both the lines involved in the intersection are visible in the current frame or only one of them has been detected.

Contrary to imposing known junctions, such intersections are identified in the 2D space. For every frame $i$ features $F_i$ are projected to the image plane and each feature $f_j \in F_i$ is tested for intersection with all other features in $F_i$. Assume the example of figure 6.3(a) where the junction between the lines $l_1,l_2$ has not been estimated. Their planar intersection at point $P_i$ is then first identified, the corresponding junction is hypothesised and the involved lines are updated with the same process described above for the known junction information case.

Figure 6.3: Example of noisy T junction between 3D lines $l_1,l_2$. (a)The image plane projections of the lines are intersected (b)Lines in 3D from a different angle.
6.2.2 Feature update and uncertainty propagation

For each new frame new measurements of existing features are integrated from SFM with the existing corresponding estimates based on a Kalman filtering update. This yields to a maximum likelihood new estimate. The methodology ensures that as the number of frames in which a feature is observed increases its corresponding covariance will decrease. Hence model features locations and variances are updated based on the corresponding new estimates computed from SFM at each frame.

Line features update to the new corresponding feature estimated orientation but the extent of the associated segment and their variance are considered after imposing the junctions as described in the previous section. A requirement that should be met from the line update process is that the new feature should be the greatest segment formed from the endpoints of the existing and new measurement so that it depicts the portion that has become visible across all the frames where it has been extracted.

Considering the example of figure 6.4 with the existing line $p = (p_1, p_2)$ and new estimate $x = (x_1, x_2)$ the steps of the update process can be summarised to:

1. Project $p_i$ onto the nearest point on the model line $(x_{pi})$.

2. If any of the $x_{pi}$ lies outside the segment $x$
   
   (a) Shift the corresponding endpoint $x_i$ so that it coincides with $x_{pi}$.

   (b) Extrapolate the covariance of new $x_i$ using equation (6.7).

This process ensures that feature positions and variances are updated based on our best current estimates for the location of the corresponding real structure.
6.3 Visibility test using feature uncertainty

The visibility of features from a specific camera position relative to the reconstructed model is a powerful tool for testing the consistency of the model. However, noise in the data can result in rejection of hypothesised triangles which actually correspond to real scene surfaces. This makes the original algorithm ‘brittle’ in the presence of noisy measurements (surfaces are incorrectly deleted). Thus visibility should be applied taking into account the uncertainty of both the model’s triangles and feature estimates.

In the reconstruction algorithm each feature \( f_j \in F_i \) in frame \( i \) defines a visibility constraint according to Definition 1. To account for the uncertainty in the measurement data, a modified definition for the visibility constraint is required:

**Definition 7 (Visibility constraint with uncertainty)**: The space between the camera position and the volumetric uncertainty envelope of the feature should not intersect with any of the model’s triangles uncertainty volumes.

This definition ensures that no violation of visibility occurs due to noise under some confidence measure. Exact implementation of the visibility tests according to the above definition involves intersections between higher order surfaces in 3-space. This is however difficult to achieve as no explicit equations exist for these surfaces. Therefore, a more conservative approach is adopted by approximating this test and reducing the search space along the direction of the ray connecting the camera centre and the feature projection onto the image plane.

The process of applying visibility for an observed 3D point \( p \) against the reconstructed model will be examined. Testing visibility for lines is a direct extension of this process for each of the two associated endpoints and evaluation of their relative depth compared to the surface under examination (section 4.3.2). Consider the example of Figure 6.5. The 3D visibility ray \( r = (c, p_t) \) originates at the camera centre \( c \) and passes through \( p_t \), the projection of \( p \) onto the image plane. Using back-projection of this ray against the plane formed by the 3D triangle \( t \), point \( p_t \) on the triangle is computed. Collinear
Figure 6.5: One dimensional visibility test. (a) Model’s triangles and features are tested for 2D overlap in the image plane. (b) Representation of 1D variance of \( \mathbf{p}, \mathbf{p}_t \) along the projection optical ray.

The problem thus reduces to the estimation of the 1D variances for both \( \mathbf{p} \) and \( \mathbf{p}_t \). The process of computing the variation envelope for \( \mathbf{p}, \mathbf{p}_t \) is similar and it will be just presented for \( \mathbf{p} \). Practically the 3D point \( \mathbf{p} \) does not lie exactly on the examined 3D ray \( \mathbf{r} \) because it is the estimate of the true position in the scene. Instead an ellipsoid centred at \( \mathbf{p} \) exists which indicates where the actual point lies in 3-space with some confidence. If this ellipsoid is considered as a cloud of possible positions for the scene feature then each of these points can be projected onto the visibility ray. Every point \( \mathbf{x} \) on the line can be expressed as \( \mathbf{x} = \mathbf{c} + \lambda \mathbf{v} \) where \( \mathbf{v} \) is the unit vector along the 3D ray.

The segment \( \mathbf{p}_v \) can be estimated by examination of the distribution of \( \lambda \) for the set of projected points. It can be shown that for each point \( \mathbf{x} \) projected, \( \lambda \) will be
6.3. Visibility test using feature uncertainty

\[ \lambda = \mathbf{v} \cdot (\mathbf{c} - \mathbf{x}) \]  

(6.11)

By considering equation (6.11) as a linear transformation of \( \mathbf{x} \) to \( \lambda \) then based on equation (6.1) the mean \( \overline{\lambda} \) and the covariance \( P_\lambda \) will be

\[ \overline{\lambda} = \mathbf{v} \cdot \mathbf{c} - \mathbf{v} \cdot \overline{\mathbf{x}} \]
\[ P_\lambda = (-\mathbf{v})^T P_x (-\mathbf{v}) \]  

(6.12)

where \( \overline{\mathbf{x}} = \mathbf{p} \) the mean of distribution of points in the ellipsoid and \( P_x \) the covariance. As it was expected from equation 6.12, \( \overline{\lambda} \) corresponds to point \( \mathbf{p} \) on the line. A confidence envelope of approximately 99% is achieved by considering a space of three standard deviations in each side of the mean. Thus points \( \mathbf{p}_v1 \) and \( \mathbf{p}_v2 \) are considered to be at distance \( 3\sqrt{P_\lambda} \) left and right of \( \mathbf{p} \) on the ray \( \mathbf{r} \). The variation envelope for \( \mathbf{p}_t \) is computed similarly.
Chapter 6. Robust Reconstruction
A framework for reconstruction of indoor scenes has been described. In this chapter the application of the developed methodology is demonstrated. Both real and synthetic sequences of scenes with multiple objects and significant occlusions are used to evaluate the reconstruction performance and validate the assumptions that have been adopted. The use of synthetic sequences enables the model reconstruction to be evaluated for varying levels of noise in the feature primitives. In a real system, noise is introduced through the lower level processes of image capture, feature extraction, matching and structure-from-motion SFM. The performance of the reconstruction process to noise and outliers is evaluated using the statistical framework presented in chapter 6.

7.1 Synthetic data

Generation of synthetic sequences is based on a platform that closely simulates the real system. An environment similar to a first person arcade game has been implemented. A 3D scene is manually designed and the operator of the experiment is able to rotate and translate the camera inside this scene in all directions. Images of the 3D world are rendered in real time as the user navigates through the virtual environment in the same way a mobile robot does. Once the reconstruction position has been picked a hidden
line detection algorithm is used to determine visibility of object faces and compute the associated visible edge sections. A file similar to the one produced by SFM is generated and fed to the modelling system. A new model is then computed using the incremental reconstruction algorithm presented in chapter 4. The user then interactively moves the camera to new locations to capture data for successive frames in a sequence for model reconstruction. New viewpoints can be selected either from a pre-defined path or manual interaction based on the users knowledge of the scene structure. Automatic viewpoint selection based on the partial reconstruction could also be used if available.

This synthesis offers several advantages relative to image sequences captured from real scenes. The controllable environment frees the reconstruction process from any complications involved in the lower levels of the system. Correspondence between 2D image features over subsequent frames is automatically known eliminating in this way the problem of feature tracking and allowing the processing of synthetic views that are widely spaced. Furthermore, prior information on scene structure and camera positions removes the dependency on obtaining data from the SFM process. This has allowed much more extensive testing of the algorithms than was possible for data available for real scenes. The ideal scenario without noise is first used to test the proposed methodology of scene reconstruction from sparse data. The performance is then evaluated under more realistic conditions with added noise and missing features at different levels. The use of synthetic data allows evaluation of the performance against known ground truth. In practice this independent evaluation was vital due to the practical difficulties experienced in obtaining reliable reconstruction for long image sequences of real indoor scenes with current automatic feature matching and structure-from-motion algorithms.

### 7.1.1 Noise free synthetic sequences

Three different synthetic worlds have been built in order to test the developed incremental reconstruction methodology. A 2D map for each of these scenes along with the approximate capturing positions picked during the incremental reconstruction process is presented in figure 7.1. An extended illustration of the corresponding rendered images for each of these scenes is presented in Appendix C.
Figure 7.1: Top view maps for the synthetic scenes. Arrows represent the approximate positions of the cameras.
Sequence 1: The first synthetic scene comprises a room with two boxes one behind the other. There are 16 images in the sequence of the camera moving around the boxes as shown in figure 7.1(a). The partially reconstructed model up to several intermediate frames from the sequence are presented in figure 7.2(a—d). The final reconstructed model is presented as a wire-frame from multiple novel viewpoints in figure 7.2(e). Analysis of the final model shows that all the triangles in the model correspond to real scene surfaces. All virtual triangles have been eliminated by the visibility constraint. Statistics for the reconstruction process are given in table 7.1. In addition, the final model is completely closed without any holes in the reconstructed surfaces. This demonstrates that the model reconstruction algorithm has converged to a valid approximation of both the true scene surface geometry and topology. The topology of the real scene has been accurately modelled despite the multiple occluded regions and the small number of images in the sequence. Notice should be taken on the compactness of the model which involves just 85 triangles to approximate the 16 surfaces in the real scene.

Sequence 2: The second sequence of twenty frames is of two empty rooms connected through a doorway. The camera moves around the first room before passing through the door and explore the second room. Again the topology has been correctly estimated as it is presented in figure 7.3. All triangles in the reconstructed model correspond to real surfaces and there are no holes. Statistics of the reconstruction are given in table 7.1. The final model consists of 111 triangles to approximate 20 scene surfaces.

Sequence 3: The third scene is of a large scale internal environment with four empty rooms connected by corridors. Starting at the centre of the scene the camera moves inside each of the four rooms exploring the whole world in 55 images. Several models corresponding to different stages in the recursive reconstruction process are presented in figure 7.4. An accurate reconstruction of the full scene surface topology is again reconstructed without holes or invalid triangles. The accurate reconstruction of the scene topology is evidence of the ability of the described methodology to reconstruct extended sequences. These results demonstrate that in the absence of measurement noise the hypothesise and verify strategy using feature visibility converges to a valid approximation of the scene with a small number of viewpoints.
7.1. Synthetic data

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Num.of surfaces</th>
<th>Number of triangle</th>
<th>Invalid triangles</th>
<th>Holes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td>85</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>111</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>78</td>
<td>455</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.1: Table of model reconstruction statistics without noise

An important feature of the algorithm is the processing time required for each frame. To study the computational performance the third sequences with four rooms has been used. Frames in the sequence have been separated to two sets. The first set involves the frames where a number of features are introduced for first time while the second frames where an estimation already exists for all the visible structure primitives. After reaching the last position the camera moves back and visits exactly the same viewpoints in the inverse order. For the reverse sequence all features have previously been seen and no new features are added.

Figure 7.5 illustrates the increase of the processing time as the number of frames extends up to around the fiftieth frame where all the structure has already become visible. However, as expected from the analysis of section 4.4 this is a linear rather than a square order increase relative to the number of frames already processed. Integrating the inverse sequence at the end of the original demonstrates the stabilisation of the process which has approximately constant time per frame. This is due to the introduction of the visibility pyramid to cull unobserved sections of the global model.

For each of the synthetic sequences that has been reconstructed exact (up to numerical precession) feature and camera position estimates have been used. These experiments may not be very realistic however they demonstrate the ability of the incremental reconstruction algorithm to produce consistent scene models for sequences of different complexity and length using only visibility constraints from the current view. These results indicates a rate of convergence much better than the theoretical proof of convergence for an infinite sequence.
Figure 7.2: Reconstructed models for the room with the boxes sequence. (a),(b),(c),(d) Models up to 1, 5, 9 and 13 images processed. (e) Final reconstructed model from three different angles.
7.1. Synthetic data

Figure 7.3: Reconstructed models for the two rooms sequence. (a), (b), (c), (d), (e) (f) Models up to 1, 5, 9 and 13 images processed. (e) Final reconstructed model from different viewing angles
Figure 7.4: Reconstructed models for the floor sequence. (a), (b), (c), (d), (e) (f) Models up to 1, 5, 9 and 13 images processed. (e) Final reconstructed model from different viewing angles.
7.1. Synthetic data

Figure 7.5: Processing time relative to the number of the frame in the sequence. Times have been measured on an 400MHz Ultra Enterprise Sun system and are rounded to the closest second

7.1.2 Synthetic noisy scenes

A more realistic evaluation than that presented in the last section involves the presence of errors in the estimated scene structure. The first two synthetic scenes have been used to access the methodology’s robustness to noisy measurements. To make the analysis quantitative details of the geometry for each of these two scenes are presented. However, as reconstruction is only up to scale these are arbitrary specified dimensions. For the first sequence the size of the room is $4 \times 3 \times 8m$, of the front box $1 \times 0.6 \times 0.4m$ and the rear box $0.8 \times 0.8 \times 0.8m$ all measured in $(x, y, z)$ coordinates. For the second scene the dimensions of each of the rooms is $4 \times 2 \times 4m$.

For each frame the exact measurements on the visible section of every line feature is available from the synthetic data. To simulate the noise in the data as delivered from the SFM process independent Gaussian noise is added to each line endpoint. The size of this additive noise is proportional to the associated lines length. Given a Gaussian random
variable $r_i = N[0, \sigma_i]$ with mean zero and standard deviation $\sigma_i$ then different levels of noise $\sigma_1 = \frac{1}{100}$, $\sigma_2 = \frac{1.5}{100}$, $\sigma_3 = \frac{2}{100}$ and $\sigma_4 = \frac{3}{100}$ are added independently in each direction on every line endpoint. Uncertainty information is simulated in the form of the corresponding covariance matrix estimate for each noisy feature. A confidence level of $3\sigma_i \approx 99\%$ is used for application of the visibility constraints in the reconstruction algorithm.

New feature estimates are generated at each frame with independent additive noise. As mentioned in section 6.2.2 the covariance of a 3D feature estimate shrinks as the number of frames where it has become visible increases. This is due to the Kalman filtering temporal integration process used by SFM. To imitate this behaviour the same process has been adopted in order to update any new measurement based on the already existing estimates. Suppose $\mathbf{x}$ is the existing estimate of a feature, $\mathbf{p}$ the new measurement after adding noise and $\mathbf{l}$ the resulting feature from merging $\mathbf{p}, \mathbf{x}$. Also assume that $P_x, P_p$ and $P_l$ are the corresponding covariances and $\bar{\mathbf{x}}, \bar{\mathbf{p}}$ and $\bar{\mathbf{l}}$ the corresponding means. Then the Kalman gain will be

$$K = P_x \ast (P_x + P_p)^{-1}$$  \hspace{1cm} (7.1)

The merged covariance and mean will then be

$$P_l = P_x - K \ast P_x$$

$$\bar{l} = \bar{x} + K \ast (\bar{p} - \bar{x})$$  \hspace{1cm} (7.2)

If $\mathbf{x}, \mathbf{p}$ are Gaussian, distributed as it has been assumed they are in section 6.1, and are independent over time then equation 7.2 is the maximum likelihood estimate with variance less than any other linear unbiased estimate [58].

To evaluate the performance of the reconstruction process under noisy conditions twenty different experiments have been performed on each noise level for both sequences 1 and 2. A different seed is selected for the Gaussian random generator process of each data set. For all experiments performed on each of the sequences the same camera capturing positions have been used as presented in figure 7.1. Figures 7.6,7.7,7.8,7.9, illustrate reconstructions of the first scene under the four different noise levels. Features are also superimposed on the corresponding images to demonstrate the effect of noise. Similar results for the second sequence of the two empty rooms are presented in figures 7.10,7.11,7.12,7.13.
7.1. Synthetic data

Figure 7.6: Reconstruction of boxes sequence with noise level $r_1$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles.
Figure 7.7: Reconstruction of boxes sequence with noise level $r_2$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0, 5, 10 frame respectively. (f),(g),(h) Final model from different viewing angles.
Figure 7.8: Reconstruction of boxes sequence with noise level $r_3$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles.
Figure 7.9: Reconstruction of boxes sequence with noise level $r_4$. (a),(b) Frames 0 and 7 with the noisy feature superimposed. (c),(d),(e) Reconstructed models up to 0,5,10 frame respectively. (f),(g),(h) Final model from different viewing angles.
Figure 7.10: Reconstruction of the two rooms scene with noise level $r_1$. (a),(b),(c) Frames 0, 6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1, 6, 15 frame respectively. (g),(h) Final model from different viewing angles.
Figure 7.11: Reconstruction of the two rooms scene with noise level $r_2$. (a),(b),(c) Frames 0, 6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1, 6, 15 frame respectively. (g),(h) Final model from different viewing angles.
Figure 7.12: Reconstruction of the two rooms scene with noise level $r_3$. (a),(b),(c) Frames 0, 6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1,6,15 frame respectively. (g),(h) Final model from different viewing angles.
Figure 7.13: Reconstruction of the two rooms scene with noise level $r_4$. (a),(b),(c) Frames 0, 6 and 14 with the noisy feature superimposed. (d),(e),(f) Reconstructed models up to 1, 6, 15 frame respectively. (g),(h) Final model from different viewing angles.
Noise in the input estimates significantly affects the geometry of the reconstructed models. This is depicted in the figures 7.14, 7.15. For each of the two scenes the mean, standard deviation and the maximum distance between the endpoints of the finally estimated structure features and the corresponding ground truth positions has been measured. Figures 7.14, 7.15 present the expected deterioration of the models as the noise increases.

The results demonstrate that the reconstructed models topology has been influenced by the introduction of noise in the feature estimates. For the first synthetic sequence with noise levels $r_1, r_2$ and $r_3$ in all experiments the scenes surfaces have correctly approximated, there are no holes and no invalid triangles as it can be seen in table 7.2. For the highest noise level $r_4$ however more than half of the reconstructions exhibit a number of holes. Despite this most of the triangles in the reconstructed model approximate real scene surfaces.

There are two main reasons resulting in holes in the reconstructed model. Consider the experiment presented in figure 7.16. Initially (figure 7.16(a)) the rear box is connected to the back wall. No junctions have been identified for any of the top corners of this back wall and thus no triangles have been hypothesised to cover this surface up to the fourth frame (figure 7.16(b)). When feature visibility invalidates the triangles that join the box with the wall in the subsequent frame the model remain with a hole (figure 7.16(c)). Due to the restrictive navigation plan the hole is not filled. It should be noted that this failure in the presence of noise levels is not catastrophic. Holes that are introduced into the reconstruction can be corrected by further image observations of the features surrounding the hole. All triangles in the final model correspond to real surfaces demonstrating the convergence to a correct partial approximation of the true surface topology in the presence of noise.

Another example where the method fails to obtain a correct approximation of the surface topology for high noise levels is illustrated in figure 7.17. The reconstructed model up to frame five is shown in figure 7.17(a). The next image is presented in figure 7.17(b) with the visible noisy measurements superimposed. In this case the big gap between the two successive views results to a very noisy estimate for the upper
Chapter 7. Application of Model Reconstruction Framework

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Noise Level</th>
<th>Surfaces</th>
<th>Average triangle number</th>
<th>Models with invalid triangles</th>
<th>Models with holes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$r_1$</td>
<td>16</td>
<td>84</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$r_2$</td>
<td>16</td>
<td>84</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$r_3$</td>
<td>16</td>
<td>84</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$r_4$</td>
<td>16</td>
<td>85</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>$r_1$</td>
<td>20</td>
<td>133</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$r_2$</td>
<td>20</td>
<td>134</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$r_3$</td>
<td>20</td>
<td>140</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>$r_4$</td>
<td>20</td>
<td>149</td>
<td>12</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 7.2: Table of model reconstruction statistics for sequences 1,2 and four different level of noise on 20 distinct experiments.

line of the side wall and thus the visibility test results in the deletion of its associated triangles on the ceiling (figure 7.17(c)). The final model is shown in figure 7.17(d). Again the high noise level does not result in catastrophic failure, all triangles in the reconstructed model correspond to real scene surfaces and the holes could be filled by capturing further images.

For the second synthetic sequence again all the experiments performed with noise of level $r_1$ and $r_2$ successfully reconstructed the scenes topology as it is presented in table 7.2. This was not achieved for noise level $r_3$ where four of the experiments resulted in a number of holes. The effect of noise aggravated on the next level ($r_4$) where most of the models involves a large number of holes. This decrease in the algorithms performance compared with the first sequence is justified from the even wider viewing angles between subsequent frames and the smaller number of frames where the same features are visible across multiple views.
Figure 7.14: Models geometry evaluation for the boxes sequence. The mean, variance and maximum distances of estimated models relative to the ground truth information.
Figure 7.15: Models geometry evaluation for the two rooms sequence. The mean, variance and maximum distances of estimated models relative to the ground truth information.
7.1. Synthetic data

Figure 7.16: Example of reconstruction failure. (a),(b),(c),(d) Reconstructed models up to frames 0, 4, 5, and 15 respectively.

Figure 7.17: Example of reconstruction failure. (a) Reconstruction up to frame 5. (b) Frame 6 with superimposed features. (c) Model after frame 6. (d) Final model
Noise is an inherent problem of any real system and in this section an attempt has been made to imitate its effect by examination of a large number of experiments on multiple scenes. For both sequences used, reliable reconstructions are accomplished for noise up to $r_3$ level. This noise level correspond to a maximum error (with 99.73% confidence) of ±6% over the line lengths. In practice this exceeds the expected noise level for estimated scene features reconstructed using structure-from-motion algorithms on real image sequences [60]. For the highest noise scale (±9%) typically multiple holes occur in the reconstructed models. This problem could be circumvented by image sequences where features are observed from different angles. This would be the case in an autonomous system where camera viewpoints are selected based on missing data in the prior global model.

The analysis of sequences in the presence of noise allows us to conclude that the reconstruction algorithm together with the statistical framework for feature updates and visibility constraints is reliable for noise levels with up to 6% error in the feature estimates. This level of error as it was demonstrated in the images where noisy features are superimposed corresponds to several pixels in the image and is therefore considerably greater than the 2-3 pixel error expected for real sequences. This section has considered the purely statistical error due to measurement noise. In the following section we evaluate the effect of other sources of error such as missing data.

### 7.1.3 Sequences with incomplete feature sets

Up until this section reconstructions have been demonstrated using a complete structure primitive set that comprises all lines and junctions visible in the scene for each camera viewpoint. Unfortunately, in a real system features are not necessarily reconstructed in all images in which they are visible. This is due to failure of the low-level processing operations such as image feature extraction, junction identification and matching across multiple frames. Therefore, visible features will be missing from the SFM reconstruction. To evaluate the model reconstruction performance for missing features, synthetic sequences with the following conditions have been tested:

1. Absence of junction information for the entire sequence.
7.1. Synthetic data

2. Absence of a random set of features from the entire sequence.

3. Random absence of a fixed proportion of features from each frame.

A set of sequences have been generated for the first scene with two boxes in the middle of a room. As in the previous section features are generated with a Gaussian feature noise of level of \( r_3 \). To facilitate the following discussion figure 7.18 presents a wire-frame view of the scene with the line features enumerated.

**Reconstruction without junction information.** The importance of junction information in the quality of the reconstructed models has been highlighted in section 6.2. To substantiate this significance the scene has been reconstructed without the use of any junction knowledge. Noise of level \( r_3 \) has also been added to the feature endpoints. The resulting triangulated scene model is illustrated in figure 7.19. Although the topology has been recovered, the model is characterised by the thin triangles that have been formed at every corner of the scene. These triangles span the space between the lines that otherwise were forming the junctions. This redundancy is reflected in the final number of triangles which is 112 and compared with the average 84 triangles for reconstructing the same scene with the same noise level an increase of 25% is observed. This result illustrates that the absence of junction information does not cause failure of the reconstruction algorithm but results in an increase in the number of triangles.
Figure 7.19: Reconstructed model with no junction information. (a),(b),(c) Final model from several viewing positions. (d) A close look at the reconstructed boxes.
7.1. Synthetic data

Figure 7.20: Reconstruction where significant line structure is missing from the input feature set. (a),(b),(c) Model viewed from different angles.

**Reconstruction with features missing from the entire sequence.** Line estimates are equivalently important to junctions especially when only a sparse feature set is available. However, a significant point is that depending on the real scene geometry and topology not all lines appear to be equally critical. To demonstrate this a number of line primitives has been randomly selected (using a uniform distribution random number generator process) and deliberately kept out of the input data during the whole sequence.

An example reconstruction of the scene where features \{3,4,24,26,30,52,65\} (almost 20% of the whole line set) are missing is presented in figure 7.20. Noise of scale $r_3$ has also has been added. The scene is correctly reconstructed except for triangles adjacent to missing features 3. The absence of this feature causes the big hole in the model. In the example of the presented synthetic scene it was found that if any of the features \{1,3,5,7\}, as referred to in figure 7.18, were missing for the entire sequence this would have resulted in holes in the model. This can be justified considering the fact that the whole extent of these lines cannot be seen in a single frame at least with the specific internal camera parameters. This evaluation demonstrates that for a scene with sparse structure detection of some features can be critical to obtaining a full reconstruction. However, in the absence of such critical features a partial reconstruction is obtained with holes which could be filled by capturing images in the neighborhood of the missing features.

**Reconstruction with features missing for part of the sequence.** The absence of features for the entire image sequence is in most practical situations overly constrain-
Chapter 7. Application of Model Reconstruction Framework

For real image sequence it is likely that features are not correctly identified or matched for part of the sequence but are correctly extracted from different viewpoints. This could be due for instance to changes in surface appearance with viewpoint. To simulate this process the initial assumption that lines remain concealed throughout the entire sequence is relaxed. Two different experiments have been conducted where a randomly selected sample of 20\% and 30\% respectively of the whole line feature set was hidden from the whole scene in each frame. All features also include noise at the \( r_3 \) level. Figures 7.21 and 7.22 present the frames from the sequences used in each case with the noisy features (level \( r_3 \)) superimposed. The associated final models are illustrated in figures 7.23 and 7.24. From this analysis with up to 20\% missing data a full reconstruction is obtained without errors. When the percentage of missing features is increased to 30\% holes appear with greater frequency and an expert navigation plan is required in order to accurately approximate the scene surfaces. This evaluation demonstrates that reliable reconstruction of surface topology can be achieved with noise levels up to 6\% and up to 20\% missing data.
7.1. Synthetic data

Figure 7.21: Sequence of frames for reconstruction when 20% of line features are hidden
Figure 7.22: Sequence of frames for reconstruction when 30\% of line features are hidden
7.1. Synthetic data

Figure 7.23: Reconstructed model from sequence with 20% of lines missing. (a),(b),(c) Models up to 2,9,16 frame respectively. (d),(e),(f) Final model.

Figure 7.24: Reconstructed model from sequence with 30% of lines missing. (a),(b),(c) Models up to 0,8,18 frame respectively. (d),(e),(f) Final model.
7.2 Real Data

Application of the surface reconstruction system proposed, integrated to the robotic mobile platform system described in chapter 1, is presented in this section. Unfortunately, due to experimental limitation, the SFM analysis was performed over the entire sequence prior to reconstruction and no control over the robot positions was available. The sequence captured involves ten images that show the corner of a lab as the robot passes by (figure 7.25).

Junction points and line features are automatically extracted and matched along consecutive frames of the acquired sequence. A recursive SFM algorithm [60] is applied and estimates for the 3D location of the sparse scene features and the camera position and orientation is computed for every frame. Our modelling algorithm was subsequently used and the resulting models at different stages of the reconstruction process are presented in figure 7.26. The topology of the real scene has been successfully approximated in the reconstructed models despite the limited number of input frames. Holes occur due to occlusion of the several objects in the scene while the lack of controlling the robot prevents the filling of gaps.

This is a limited experiment but provides some evidence on the ability of our reconstruction algorithm to produce a reasonable model for images of a real scene. Further evaluation for the reconstruction of real scenes is however required before definitive conclusions can be drawn. This was not possible due to failures of the structure from motion process on several sequences of indoor scenes captured for this purpose. Failure occurred due to insufficient feature matching information between consecutive frames to constrain the projective reconstruction. Therefore, the simulation system for generating synthetic structure-from-motion sequences described earlier in this chapter was developed.
7.2. Real Data

Figure 7.25: Real sequence from the corner of a lab

Figure 7.26: Reconstruction of the real scene sequence. (a),(b),(c) Models up to 4, 6 and 9 frame
Chapter 8

Discussion, Conclusions and Further Work

Development of an autonomous vehicle to navigate an unknown environment and reconstruct full 3D scene models has been a long term goal in the field of machine vision. An ambitious system has been designed to target this problem by using only monocular video input. In contrast with the recent trend of integrating multiple sophisticated sensors (LRS, ultrasound, GPS) the aim of this work was to investigate if it is possible to reconstruct environment models using only passive visual sensing of video image sequences. The use of a simple camera has potential advantages in terms of cost and simplicity of integration.

The research presented in this thesis has primarily focused on the sub-problem of reconstruction of a scene surface model given a sparse set of 3D scene features estimates and camera locations together with their covariances. A theoretical analysis of the problem of reconstruction from sparse feature estimates in N-views has been presented. This analysis has led to the development of a general solution to the problem of reconstructing a scene model, which is a planar approximation of an arbitrary scene using feature visibility constraints. This algorithm has been shown to converge to an approximation of the real scene surfaces as the number of views increases. An incremental algorithm has been introduced which gives a computationally efficient approximation to the general algorithm and converges in the limit to a valid approximation of the real
scene. Implementation of this algorithm and subsequent evaluation on synthetic and real sequences has enabled the following conclusions to be reached:

- The incremental reconstruction algorithm using feature visibility allows correct reconstruction of scene surface topology and geometry for indoor environments of moderate complexity.
- The algorithm using feature visibility converges rapidly to only contain triangles which approximate the true scene surface and eliminate false surface hypothesis. The actual number of views required will depend on the particular scene structure.
- The computational cost of the algorithm is approximately linear in the number of scene features and views.
- Reliable scene reconstruction is achieved in the presence of noise using feature covariance estimates in applying feature visibility.

Several achievements have been accomplished in the field of indoor scene reconstruction from sparse feature sets [57, 55, 62, 56]. The sparseness of the input structure primitive set is the central point of this study and differs from previous work on 3D reconstruction using dense range maps in that the local scene topology cannot be estimated directly from a single view (section 3.1).

In chapter 3 a novel theoretical analysis of the problem of reconstructing surface models from sparse 3D data has been presented. This analysis has resulted in an algorithm that provably converges to the real scene topology as the number of views captured is increases. The basis for this algorithm is the use of feature visibility. No prior assumptions about the structure of the scene are required except that no transparent surfaces are existing. This N-view reconstruction methodology is based on a hypothesise and verify strategy and it is independent of the order that frames are processed.

A practical problem with the general framework is that although the model is built incrementally for each frame the algorithm relies on a history of feature visibility information from all prior views. This results in a computational cost of $O(N^2)$ where $N$ is the number of frames in the sequence. To circumvent this problem, the order...
independence characteristic is relaxed. An algorithm that achieves an overall linear complexity, \(O(N)\), over the number of frames and so ensures efficient applicability on extended video sequences has been presented in chapter 4. The algorithm takes as input features (junction points and lines) and it is shown to yield models that converge to an approximation of the real scene surface topology.

In the subsequent chapter 5 a new planar constrained Delaunay triangulation algorithm was introduced. This algorithm is incremental in the sense of stepwise addition of new features while maintaining a Delaunay triangular model. The algorithm is used in the scene model reconstruction to efficiently introduce newly observed features into the global model. The algorithm is shown to exhibit an asymptotically optimal performance.

The final contribution of this research involves a statistical framework that makes the reconstruction process tolerant to noisy measurements and it is presented in chapter 6. The covariances associated with the estimates of both feature positions and camera parameters for each frame are utilised to build an uncertainty envelope around the reconstructed model. This envelope prevents invalid deletion of modelled structure during the verification stage of the algorithm and helps make the whole process more robust.

The applicability of the model has been extensively evaluated on a series of synthetic sequences of 3D feature estimates. Repeated tests have been performed for sequences with missing data and added noise at multiple levels. The results obtained from this analysis demonstrate that the proposed hypothesise and verify strategy using feature visibility yields models that closely approximate the actual scenes topology. Evaluation of the statistical framework has been performed by introducing different levels of noise in the estimated feature measurements. The robustness of the model reconstruction process has been demonstrated for a scale of noise greater than that expected from a real SFM system. Finally, despite the already sparse form of the feature set the method manages to yield accurate models even under incomplete input structure primitive sets.

A limited evaluation has been performed on feature estimates obtained from real image sequences. This demonstrates reasonable reconstruction of the real scene surfaces.
Evaluation on data from real scenes has been limited due to difficulties experienced in obtaining structure-from-motion estimates for indoor scenes with sparse features. Although the reconstruction results from the limited real set available is accurate and plausible, more extensive assessment is required. Unfortunately, due to problems in the recursive SFM process no other set has been provided and access on other similar results proved difficult to obtain.

A number of publications have resulted directly from the work on model reconstruction from sparse feature data:


In addition, the complete system and mobile robot platform as presented in Chapter 1 has been published in:


8.1 What is the next step?

Despite the significant advances made by the research presented in this thesis towards model reconstruction from sparse data, this area of research is by no means finished. The most important limitation of this study is the absence of a comprehensive evaluation on real data.
A further verification stage could be introduced into the sparse reconstruction algorithm based on matching the appearance of hypothesised surface regions across multiple views. Ensuring that hypothesised surface regions have the same appearance from all views in which they are visible (allowing for illumination and surface reflectance) would provide a further surface visibility constraint to verify the results of feature visibility and potentially make the process more robust and faster to converge to the real scene topology.

Maybe the most important limitation of the developed algorithm is its applicability to planar scenes that can only contain planar objects. This possibly is not very restricted for indoor scenes and basically man made environments that are mostly structured with planar faces but certainly narrows the general utilisation of the process. Nevertheless in case 3D estimates of features lying on curved objects exist, a triangular approximation of the curved surface can be constructed as illustrated in Appendix 4. The geometric reconstruction theory introduced in chapter 3 is general enough for incorporating curve features in the process. However, curves have not been considered in the possible estimated feature set of the overall system and thus no provision has been made for such primitive in the developed algorithm. Curves and curved surfaces can be reconstructed based on a piece-wise planar approximation. However, the resulting representation is unlikely to be efficient. Extension of the theoretical analysis to the problem of general curved feature visibility is required to provide a satisfactory solution.

Further work is also required in the area of visualising the reconstructed models. The flat shaded VRML models currently produced may sometimes be misleading and thus texture from the video images should be superimposed on them in order to achieve realism.
Chapter 8. Discussion, Conclusions and Further Work
A theory for N-view reconstruction of a consistent triangulated 3D model has been presented in chapter 3. The approach is based on an incremental integration of new local models to an existing globally consistent representation and has been proved to converge to the true scene topology. An important characteristic of this methodology is that the order in which views are processed does not affect the final reconstruction.

To prove this order independency property consider the integration process between two views $i$ and $j$ as a binary operator (denoted with $\bullet$) applied to the corresponding local consistent triangular models $M_i, M_j$ which results in $M = \{M_i \cup M_j\}$. Also assume that there is a set of visibility constraints $C = \{C_i, C_j, C_{i,j}\}$, $C_{i,j} = C_i \cup C_j$, associated to features visible only in $i$, only in $j$ or both $i, j$ views correspondingly. First it can be shown that the operator is commutative such that:

$$M_i \bullet M_j = M_j \bullet M_i \quad (A.1)$$

Denoting $\{M_i^c\}_{c \in C; i \in \{i,j\}}$ as the subset of $M_i$ that is invalid to $c$ visibility constraints and applying steps 4, 5, 6 of the reconstruction algorithm for integrating $j$ to $i$ and inversely we get
Appendix A. Order independency of reconstruction theory

$$[M_i - M_i^{Cj}] \cdot [M_j - M_j^{Ci}] = M_i' \bigcup M_j' \quad \text{(A.2)}$$

$$[M_j - M_j^{Ci}] \cdot [M_i - M_i^{Cj}] = M_j' \bigcup M_i' \quad \text{(A.3)}$$

As the union of two sets is commutative, relation A.1 is valid. Using similar reasoning
it can be proved that A.1 stands even if one of $M_i, M_j$ is a global model consistent with
visibility constraints over a set of views.

A second important property of the methodology is that is associative. Assuming a
third view $k$ and its corresponding triangular set $M_k$ it will be proved that

$$(M_i \cdot M_j) \cdot M_k = M_i \cdot (M_j \cdot M_k) \quad \text{(A.4)}$$

Relation A.4 indicates that integrating a new view $k$ to an existing consisten pair
of views $i, j$ is equivalent to integrating the initial view model $i$ to a model con-
sistent with $j, k$. Note that the order of views inside the parenthesis is not im-
portant according to A.1. The set of visibility constrains set now becomes $C =$
$\{C_i, C_j, C_k, C_{i,j}, C_{i,k}, C_{j,k}, C_{i,j,k}\}$ and the sides of A.4 can be written as

$$(M_i \cdot M_j) \cdot M_k = ([M_i - M_i^{Cj} - M_i^{Cj,k}] \bigcup [M_j - M_j^{Ci} - M_j^{Ci,k}]) \cdot M_k$$
$$= [M_i - M_i^{Cj} - M_i^{Cj,k} - M_i^{Cj,k}] \bigcup [M_j - M_j^{Ci} - M_j^{Ci,k} - M_j^{Ci,k}] \bigcup$$
$$[M_k - M_k^{Ci} - M_k^{Cj} - M_k^{Cj,i}]$$

(A.5)

$$M_i \cdot (M_j \cdot M_k) = (M_j \cdot M_k) \cdot M_i$$

$$= ([M_j - M_j^{Ck} - M_j^{Ck,i}] \bigcup [M_k - M_k^{Cj} - M_k^{Cj,i}]) \cdot M_i$$
$$= [M_j - M_j^{Ck} - M_j^{Ck,i} - M_j^{Ck,i}] \bigcup [M_k - M_k^{Cj} - M_k^{Cj,i} - M_k^{Cj,i}] \bigcup$$
$$[M_i - M_i^{Cj} - M_i^{Cj,k} - M_i^{Cj,k}]$$

(A.6)
Comparing equations A.5 and A.6 proves that our recursive reconstruction approach is associative. Again any of $M_i, M_j, M_k$ can be considered as global models over a set of views.

Using the commutative and associative properties that has been proven to characterise the methodology any chosen order of processing views can be shown to be equivalent to any other. Therefore the theoretical algorithm presented in chapter 3 is order independent.
Appendix B

Example application of Constrained Delaunay Triangulation algorithm

Estimation of the surfaces that span the space between the cloud of 3D features estimated from SFM is achieved using a hypothesise and verify strategy. The surface hypothesis step is based on the planar Constrained Delaunay Triangulation over the projected scene primitives as they have been seen from a specific viewpoint. A new incremental algorithm presented in chapter 5 has been developed to yield such a planar triangular subdivision. In this appendix the algorithm is described as it is applied on example meshes for imposing point and line constrains.

B.1 Point insertion example

Adding a new point \( p \) to an existing triangulation \( T \) is illustrated in figure B.1. Initially the triangle \( t \) bounding \( p \) is identified (figure B.1(a)). Point \( p \) is then connected to the vertices of \( t \) (figureB.1(b)) forming three new triangles \( t_1, t_2, t_3 \) and \( t \) is removed. The three new edges are all Delaunay [49] while \( e_1, e_2, e_3 \) are pushed in the stack in order to inspect if they still satisfy the empty circle criterion (figureB.1(c)). Edges \( e_3, e_2 \) are
Appendix B. Example application of Constrained Delaunay Triangulation algorithm

tested sequentially and as \( p \) lays outside their corresponding circle they are Delaunay edges and remain unchanged. However, for edge \( e_1 \) the circumscribed circle of triangle \( t' \) contains \( p \) (figureB.1(d)). The edge thus is swapped resulted in two new triangles \( t'_1, t'_2 \) (figureB.1(e)) with edges \( a, b \) are inserted into the stack. Both the new triangles are tested and as the are valid the algorithm terminates (figureB.1(f)).
Figure B.1: Point insertion example. Lines in the stack have been sketched with dots.
Appendix B. Example application of Constrained Delaunay Triangulation algorithm

B.2 Line segment insertion example

An example of imposing a line segment to an existing mesh is presented in figure B.2. It has been assumed that its endpoints $A, G$ has already been added using the point insertion algorithm and its bounding polygon is $ABCDEFGHIJK$. The intersected edges are added to the list sorted in order of intersection along $AG$. The initial bounding polygon, the intersected edges and the constraint is illustrated in figure B.2(a). For the succeeding description an edge is referred to as convex or concave depending on the quadrilateral formed by the two adjacent triangles having this edge common.

The first edge to be processed is $BK$ and as it is convex it swaps to $AJ$. Both edges $AK, KJ$ are Delaunay edges and do not have to be tested. The subsequent edge $BJ$ is concave and thus pushed onto the stack (figure B.2(b)). In figure B.2(c) the swap of the next edge $CJ$ to $BD$ is presented. Edge $BJ$ at the top of the stack is still concave and so remains there. The following edge to be processed is $DJ$ which is swapped to $BI$ (figure B.2(d)). Again $BJ$ is checked and as it has become convex it swaps to $AI$. Edge $BI$ intersects the constraint $AG$ and so is inserted onto the stack (figure B.2(e)).

In the list of unprocessed edges the succeeding one is $DI$ which is convex and it swaps to $BE$ (figure B.2(f)). This results in edge $BD$ being tested using the Delaunay criterion. As the circle passing through points $B, C, D$ (figure B.2(g)) encompassing point $E$ the edge is invalid and so swaps to $CE$. Edge $BI$ on the top of the stack becomes convex as well and thus swaps to $AE$ (figure B.2(h)).

The next edge is $EI$ which swaps to $AH$ (figure B.2(i)) and edge $AI$ is then checked with the Delaunay criterion. The circle circumscribing triangle $AJI$ encloses $H$ as illustrated in figure B.2(j)) and so $AI$ swaps to $JH$. The subsequent edge to be processed is $EH$ which is convex and so swaps to $AF$ (figure B.2(k)). Edge $AE$ then violates the Delaunay constraint and is thus also swapped to $BF$ as can be seen in figure B.2(l).

Final intersected edge is $FH$. This edge is convex and its swap results in the original constraint $AG$ (figure B.2(m)). This process though invalidates edge $AF$ (figure B.2(n)) which swaps to $BG$ and this in turn results to the swap of $BF$ to $EF$ (figure B.2(o)). The resulting triangulation has constraint $AG$ imposed and all triangles satisfy the Delaunay empty circum-circle constraint.
Figure B.2: Line segment insertion example. Filled lines correspond to swapped edges, dashed lines to the edges in the list waiting to be processed and dotted lines to edges in the stack.
Appendix B. Example application of Constrained Delaunay Triangulation algorithm
Appendix C

Synthetic Sequences

An environment very similar to a first person game has been constructed in order to closely imitate the way the real system with a single camera is navigated around the scene and captures snapshots of its structure from several viewpoints. Three different scenes has be constructed. The first consists a room with a couple of boxes in the middle. The second, little more complicated, presents two empty rooms connected through a doorway while the last is the longer and involves a whole floor with four rooms and corridors. All three scenes are simply textured and lighted so that the bounds between the faces of the participated objects are clearly distinguishable.

In this appendix and for each of the three sequences, images of the corresponding scenes are illustrated (figures C.1,C.2,C.3,C.4), as rendered from the viewpoints that the operator chooses during the incremental reconstruction process. The first scene consists of 16 images the second 20 and the third and final 55 images.
Figure C.1: Frames from the room with boxes synthetic sequence
Figure C.2: Frames from the 2 empty rooms sequence
Appendix C. Synthetic Sequences

Figure C.3: Frames from the floor sequence (0-25)
Figure C.4: Frames from the floor sequence (26-54)
Appendix D

Reconstruction of Curved Surfaces

The system presented in this study addresses the problem of reconstructing a consistent model of an arbitrary planar scene without any prior information on its topology. For indoor environments that consists primarily of man made objects this planarity restriction does not impose a serious limitation. Nevertheless, curved surfaces will always exist in a real scene. In case the corresponding curved object is flat shaded there is an inherent limitation in estimating its 3D geometry using a SFM process as no reliable matching and tracking of features on its surface can be performed.

Assume however that the object is textured in such a way that the 3D position of features can be estimated on its surface. Then its topology can be approximated to a precision determined by the density and location of this feature set by a triangular mesh. To demonstrated this a new synthetic sequence similar to the one with the two boxes in the middle of a room has been constructed where a cylinder approximated by a dodecahedron has been placed on top of the rear box. All thirty edges of the dodecahedron are assumed visible as the camera moved around the room. The captured images are illustrated in figure D.1. Using the described methodology a 3D model of the scene is progressively built. The reconstruction of this model up to different frame instances is presented in figures D.2(a),(b),(c),(d). The final model along with some close views of the cylinder are finally shown in figures D.2(e),(f),(g).
Figure D.1: Frames from the room with the cylinder sequence
Figure D.2: Reconstruction of the cylinder sequence. (a),(b),(c),(d) Models up to 0,4,7 and 10 frame respectively. (e),(f),(g) Final model and close views of the cylinder.
Bibliography


Bibliography


