DELAUNAY TRIANGULATION BASED SPARSE 3D RECONSTRUCTION OF INDOOR SCENES USING A MONOCULAR CAMERA

by

Faisal Bukhari

A research study submitted in partial fulfillment of the requirements for the degree of Master of Science in
Computer Science

Examination Committee: Dr. Matthew N. Dailey (Chairperson)
Dr. Nitin V. Afzulpurkar (Member)
Dr. Sumanta Guha (Member)

Nationality: Pakistani
Previous Degree: Master of Computer Science
Punjab University College of Information Technology,
Lahore, Pakistan

Scholarship Donor: Higher Education Commission (HEC), Pakistan - AIT Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
May 2009

Acknowledgment

I am grateful to ALLAH, THE ALMIGHTY for giving me the strength and wisdom to complete this research in time. When I entered AIT in August 2007, I have some serious shortcomings particularly in programming. I had studied lot of programming courses previously, but due to some unavoidable circumstances, I have never joined a software company, in order to enhance my skills further. I felt that I was lacking something in my tool kit. Then I took the WAE (Web Application Engineering) course with Matthew N. Dailey. I immediately realized that he is the right teacher for doing research at AIT. I gathered information about his area of interest and then started doing coursework in his domain of interest. He helped and supported me a lot in my research. He has been very friendly and supportive. He has always given me regular time, around 2 hours per week. He has stayed in contact with me (on Google talk, SVN and email). He has checked my write up several times and has given rapid feedback. He has also guided me in selecting courses relevant to my research. I am sure that I could not do 3D reconstruction of indoor scenes without his supervision. I am confident that some part of this research, will lead towards my Ph.D., in the near future. Now I am at ease with the theoretical foundations of this topic and my programming skills. He has performed the role of a mentor. I am also thankful to Nitin V. Afzulpurkar for his cooperation and useful comments during my presentations and Sumanta Guha for accepting my request of being a part of my examination committee. My special thanks to Ran Zask for his code, dataset and guidance and Suwan Tongphu for providing code for texture mapping. I am also thankful to Irshad Ali and Sheikh Zia Uddin for their cooperation and support during this research study.

Abstract

The role of robots is increasing in our daily life. They can wash dishes, explore unknown terrain, perform security tasks, assist surgeons in patients' operations, perform surveillance and many more. Human beings gather information about the terrain using eyes and motion of the body. In other words, they see the world in 3D and accumulate information about the structure of the world. In order to approximate some of the capabilities of human beings in robots, we need the ability to create 3D models of the scene, so that robots can navigate smoothly. This technology could be utilized by armed forces, in order to understand the structure of a collapsed building to locate positions of terrorists, or by rescue robots, for example, to understand the structure of a building to locate survivors. In this research study, I make a 3D model of an indoor scene using structure from motion. My method is based on sparse points from a pair of images. My 3D model is based on 2D Delaunay triangulation. I choose this triangulation, because it gives maximum weight to the least angle of all the angles in a triangle. This triangulation usually avoid skinny triangles and is thus most suitable for mapping image texture to a 3D mesh. I render the 3D model using OpenGL.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

There is a rapidly growing body of research in the field of computer vision over the last few years. This is due to the natural inclination of the researchers and growing demand from industry for this field. One of the hottest areas in computer vision is the automatic reconstruction of three dimensional (3D) scenes. Some of the benefits include surveillance, automatic navigation of robots and vehicles, computer graphics, virtual reality, and remote surgery. Cyganek and Siebert (2009), say that the natural question often arises in our mind, what is the difference between taking one picture of a scene versus taking more than one picture of the same scene with different viewpoints. The answer is obvious, that taking two or more images of the scene from different viewpoints helps us to obtain depth information by means of geometry. 3D information is obtained through a process known as triangulation. This is why having two eyes help us. No one can deny the importance of two-dimensional (2D) maps. But, Yanco and Drury (2004), say that during human-robot interaction in urban search and rescue (USAR), operators spend an average of 30% of their time solely in situation awareness (SA) activities. Thier job is to guide the robots in unknown terrain for naviagation, but their significant amount of time was wasted in SA. By SA, they mean the efforts of operators to understand robot's location, surroundings and their current status at point of time. Nonato, Minghim, Oliveria, and Tavares (2002) say that a primary visualization problem common to many applications is the construction of 3D models from 2D data. For example, 2D information collected from different image acquisition devices may be used to obtain 3D models showing internal body strutures. These 3D models are very benefical to doctors during surgery and diagnosis.

1.2 Problem Statement

The problem of 3D reconstruction and modeling has been discussed extensively in the literature, but it is still not in the list of solved problems. The ultimate goal for most applications would be to design more autonomous systems that require less human intervention. This research is aimed to help human operators with better visualization of 3D scenes, so that they can teleoperate with more accuracy in unknown environments. There are several techniques to construct 3D models in the field of computer vision, some of the which are:

- Structure from motion (Pollefeys et al., 2004; Johnson & Kang, 1999; "Iso09", n.d.-a).
- Monocular vision (Saxena et al., 2007; "Dyn06", n.d.-b; Saxena et al., 2008).
- Image-based rendering (Uyttendaele et al., 2004).
- Structured lighting (Moravec, 1996).
- MonoSLAM (Davison et al., 2007).

1.3 Objectives

In this research study, I focus on the problem of inferring 3D structure using structure from motion technique on monocular images. The primary goal of this research is to create a 3D model of indoor environments. By attaining this goal, I will help human operators to assist robots in an efficient way during their navigation in indoor structures (buildings, airports, hotels, etc). The operators can visualize the 3D models of the routes of the robots and provide accurate guidelines to the robots. In order to achieve this goal, I will focus on the following specific objectives:

1. Construct a 3D model from sparse points using a pair of images from a single camera.
2. Use Delaunay triangulations to create the 3D model.

1.4 Assumptions and Limitations

1. The 3D model is designed only for indoor scenes.
2. The baseline distance is not too small, as this would introduce large triangulation error, for those points which are away from the camera.
3. The images are not taken too far apart, otherwise, they would be too small and finding correspondences between them become very difficult.
4. The pair of images are not from uniform looking structures; otherwise, finding correspondences become a difficult task.
5. Only the 3D points, appear in two images will be used.

1.5 Research Study Outline

Chapter 2 reviews the literature related to structure from motion (SFM), monocular vision, structured lighting, image-based rendering, MonoSLAM, and other relevant topics.

Chapter 3 describes the methodology Iused for this research, including a system overview, system design, and implementation.

Chapter 4 describes the experiments I performed, my results, and the problems I faced at each step.

Chapter 5 concludes the research, summarizes the contribution, and outlines future work.

Chapter 2

Literature Review

2.1 Background

This chapter discusses some of the important literature relevant to the reconstruction of 3D models. There are several techniques to construct 3D models in the field of computer vision. I will discuss some of the important methods, namely: structure from motion (SFM), monocular vision, structured lighting, image-based rendering, and MonoSLAM. I also discuss related topics in this chapter.

2.2 Importance of 3D Maps for Human Operators, for Controlling the Robots

Yanco and Drury (2004) say that in urban search and rescue (USAR) with human-robot interaction, it is the responsibility of robots to provide situation awareness (SA) to the operators, especially since the USAR operators are usually unaware of the robots' surrounding terrain. They found that human operators spend an average of 30% of their time on situation awareness (SA) activities. Thier job is to guide the robots in unknown terrain for naviagation, but their significant amount of time was wasted in SA. By SA, they mean the efforts of operators to understand robot's location, surroundings and their current status at point of time. Human operator, lacks 3D maps, and cannot estimate the space behind the robot, in front, or on the sides of a typical robot. If operators had 3D maps, they could get better SA, and could help the moving robot, to efficiently navigate to the desire points

2.3 Books on SFM

The book, Multiple View Geometry in Computer Vision by Hartley and Zisserman (2004) is said to be a bible on SFM. Some of the other good books with SFM material are:

- *An Introduction to 3D Computer Vision Techniques and Algorithms* by Cyganek and Siebert (2009).
- *Computer Vision A Modern Approach* by Forsyth and Ponce (2003).
- *Introductory Techniques for 3-D Computer Vision* by Trucco and Verri (1998).
- *An Invitation to 3-D Vision* by Ma, Soatto, Kosecka, and Sastry (2004).

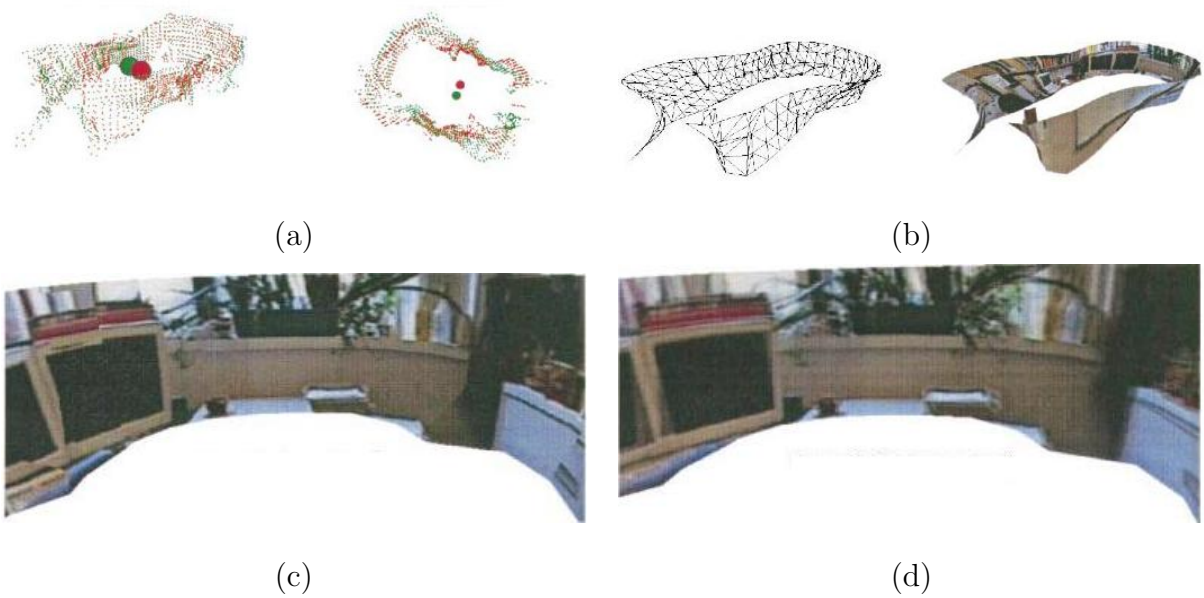


Figure 2.1: The result of combining two sets of 3D textured points of an office. (a) Registered points. (b) Wire frame surface and texture. (c-d) Closeups of textured models using different blending methods. Reprinted from Johnson and Kang (1999).

2.4 Structure from Motion

Johnson and Kang (1999) pointed out that multiple views are needed in order to create 3D models of complex object or indoor scenes. They discuss a methodology for merging 3D textured data sets, each of which is derived from different view points for an indoor scene. The merging process consists of two steps:

- Registration.
- Integration.

The registration process is based on a modified version of the iterative closest point (ICP) algorithm, which they name a color ICP. It relies not only on 3D information but also on color. When using omnidirectional stereo data, color significantly reduces registration error. But there is a danger of accumulating more local minima with color, as it is obviously a function of both the shape and texture distribution. This problem can be eradicated by adding a simulated annealing-like characteristics to the algorithm. After registering all of the 3D data sets, they integrate them to a produce consistent 3D textured model. This registration process is based on 3D occupancy grid, which is constructed through a voting process. The authors, also store surface normals in each voxel of the occupancy grid; the normals are used identify the surface from the occupancy grid. From the obtained surface they attach textures from multiple views.

Pollefeys et al. (2004) have developed a complete visual model from camera images. Their system is suitable for an uncalibrated image sequence captured from a hand-held camera. If k, the calibration matrix is known, then the camera is said to be calibrated; otherwise

it is uncalibrated. The relationship between multiple views are calculated on the basis of tracked or matched features. They thus obtain structure from the scene and motion of the camera. The projective structure and motion is transform to a metric reconstruction using self calibration. The results are further improved using Bundle Adjustment (BA), which is a nonlinear method to minimizing reprojection error. Then images are rectified. A dense stereo algorithm is used to match images and a dense accurate map is obtained by accumulating measurements from multiple images. Different types of visual models can be built such as a: 3D textured mesh, an image based approach, and view dependent geometry and texture. The last approach gives their system the capability to capture complex scenes. Their method applied to a video sequence or a sequence of still images.

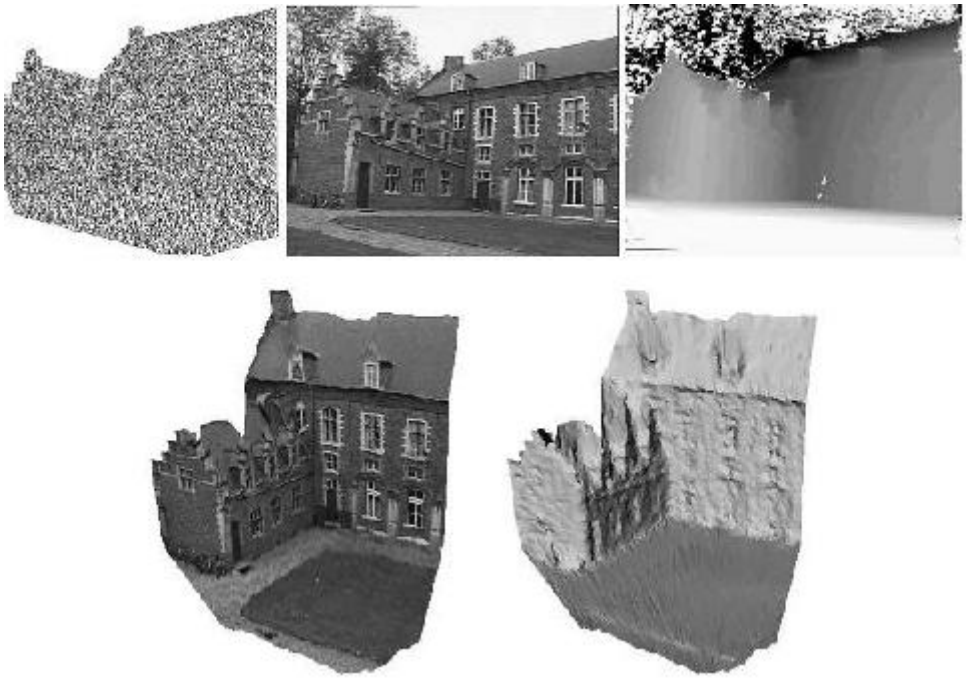


Figure 2.2: The image is covered with a triangular mesh. The sets of 2D points are converted to a set of 3D points to generate the 3D model. Reprinted from Pollefeys et al. (2004).

Schreiber, Li, and Baker (2008) have reanalyzed the method of Tsai, Huang, and Zhu (1984) for the computation of camera related parameters, especially camera motion parameters. They have point out some of the shortcomings and computational failures that are inherent in Tsai and Huang (1984). They recommend new formulas that avoid these problems.

"Iso09" (n.d.-a) present an algorithm for 3D modeling, which is incremental in nature, and according to them, suitable for the real time implementation in an indoor environment. Their algorithm constructs a texture based polygonal mesh model of an indoor scene using a monocular camera based video sequence of images. By the introduction of new images, the isosurface of a coarse 3D occupancy grid is changed incrementally. It provides low metric accuracy but on the other hand, it assists in filtering the measurement noise and also helpful for rapid 3D reconstruction (Figure 2.3). They show that if the requirement for photorealistic rendering is relaxed, then it is possible to get the benefits of both structure from motion and SLAM. They obtain roughly accurate 3D model, which is enough for human operators to guide robots in an efficient way. There are two primary ways to build surface models, one

is from 3D points and other from volumetric data. The first method uses triangulation of 3D points to get a polygonal mesh. The second main alternative method utilizes the concept of sensor model to update an occupancy grid as new images arrive. Occupancy grids define environment as a set of cells that show the probability of occupancy by some obstacle. The 2D occupancy grid is commonly used with sonar and laser sensors for localization, mapping, and navigation, and the 3D grid is a straightforward generalization of the 2D grid. Their 3D model suits search and rescue problems well. But they have not recommended it for any other specific application. Figure 2.4 shows the isosurface-based textures 3D model.

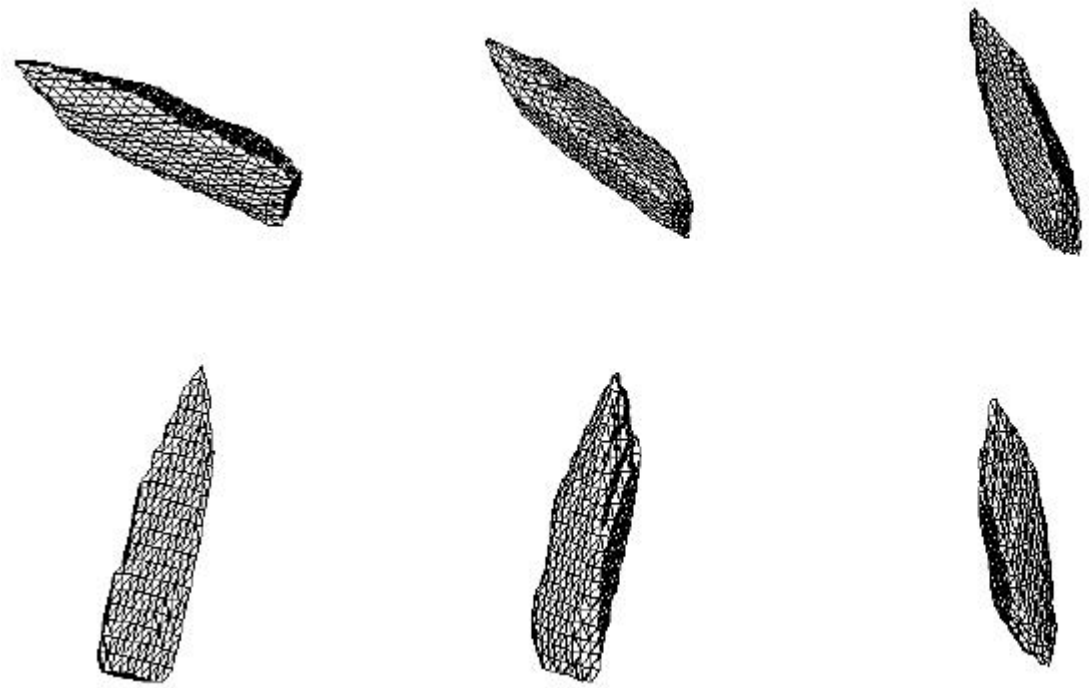


Figure 2.3: Isosurface-based 3D model. Reprinted from "Iso09" (n.d.-a).

2.5 3D Reconstruction Using a Single Image or Few Images

"Dyn06" (n.d.-b) present a dynamic Bayesian network (DBN) model (Figure 2.5 and Figure 2.6), which is a representation of a sequence of random variables; usually DBNs represent time series. They have focused on prior knowledge, which helps them to recover distances from a single image It is capable of solving some of the problems associated with monocular vision and creating 3D information for many images. They assume a floor-wall boundary in the scene and make their model; capable of recognizing that floor-wall boundary in each column of the image. Using their approach, a 3D model can be generated from a single image, when it is under perspective geometry. They claim this is a pioneering approach to constructing 3D models using single indoor images.

Saxena, Sun, and Ng (2007) construct 3Ds model of big scenes, using a small number of images of the scene. It is a difficult problem due to the following reasons:

Figure 2.4: 3D Model mapped with textures from a single image. Reprinted from "Iso09" (n.d.-a).

- The images are taken from various viewpoints.
- If the images contain repeated looking structure, then most geometric reconstruction methods will have difficulty finding good correspondences.
- Most reconstruction methods fail for registers of the scene only imaged once.

Their algorithm makes a strict assumption that the environment is composed of a small number of planes. Previously, most 3D models constructed using a single image, will be only partially complete. They show how monocular image cues can be amalgamated with triangulation cues to build a close-to-real model. This model is based only on a few images from different perspectives. Each image is segmented into small patches called superpixels. Then for each image, they concurrently try to estimate the 3D position and orientation of each small patch. A Markov Random Field (MRF) is applied for this purpose. The algorithm is the first in the field, that has the potential automatically build full photo-realistic 3D models using few images (Figure 2.8).

In more recent work from the same lab, Saxena, Chung, and Ng (2008) have opted to supervised learning approach to 3D depth estimation. They gathered a dataset of monocular images for unstructured indoor and outdoor scenes and corresponding ground-truth depth maps. The model is based on a hierarchical, multiscale Markov Random Field (MRF). Their proposed model is based on both monocular cues and stereo cues (Figure 2.9)

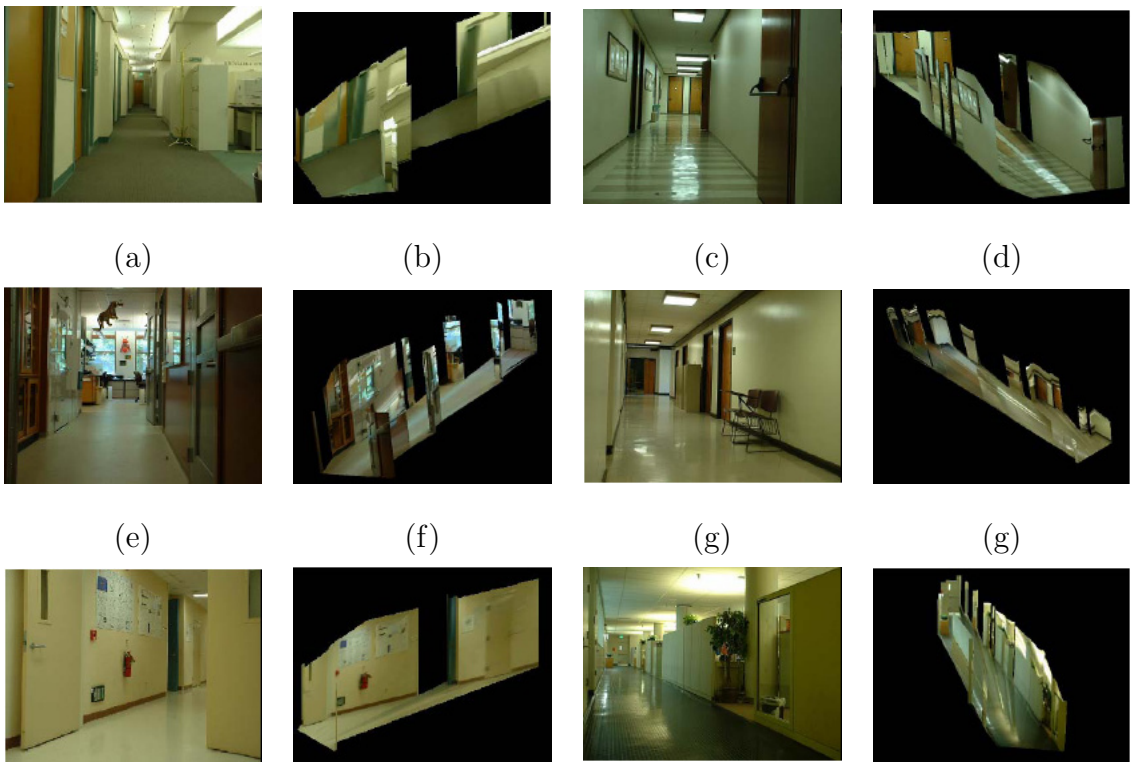


Figure 2.5: A dynamic Bayesian network (DBN) based 3D models. (a-b) represent original image and the constructed 3D model of the indoor scene using DBN. (c-d) represent original image and constructed 3D model of the indoor scene using DBN. (e-f) represent original image and the constructed 3D model of the indoor scene using DBN. (g-h) represent original image and the constructed 3D model of the indoor scene using DBN. (i-j) represent original image and the constructed 3D model of the indoor scene using DBN. (k-l) represent original image and the constructed 3D model of the indoor scene using DBN. The images are obtained from the Internet. Reprinted from "Dyn06" (n.d.-b).

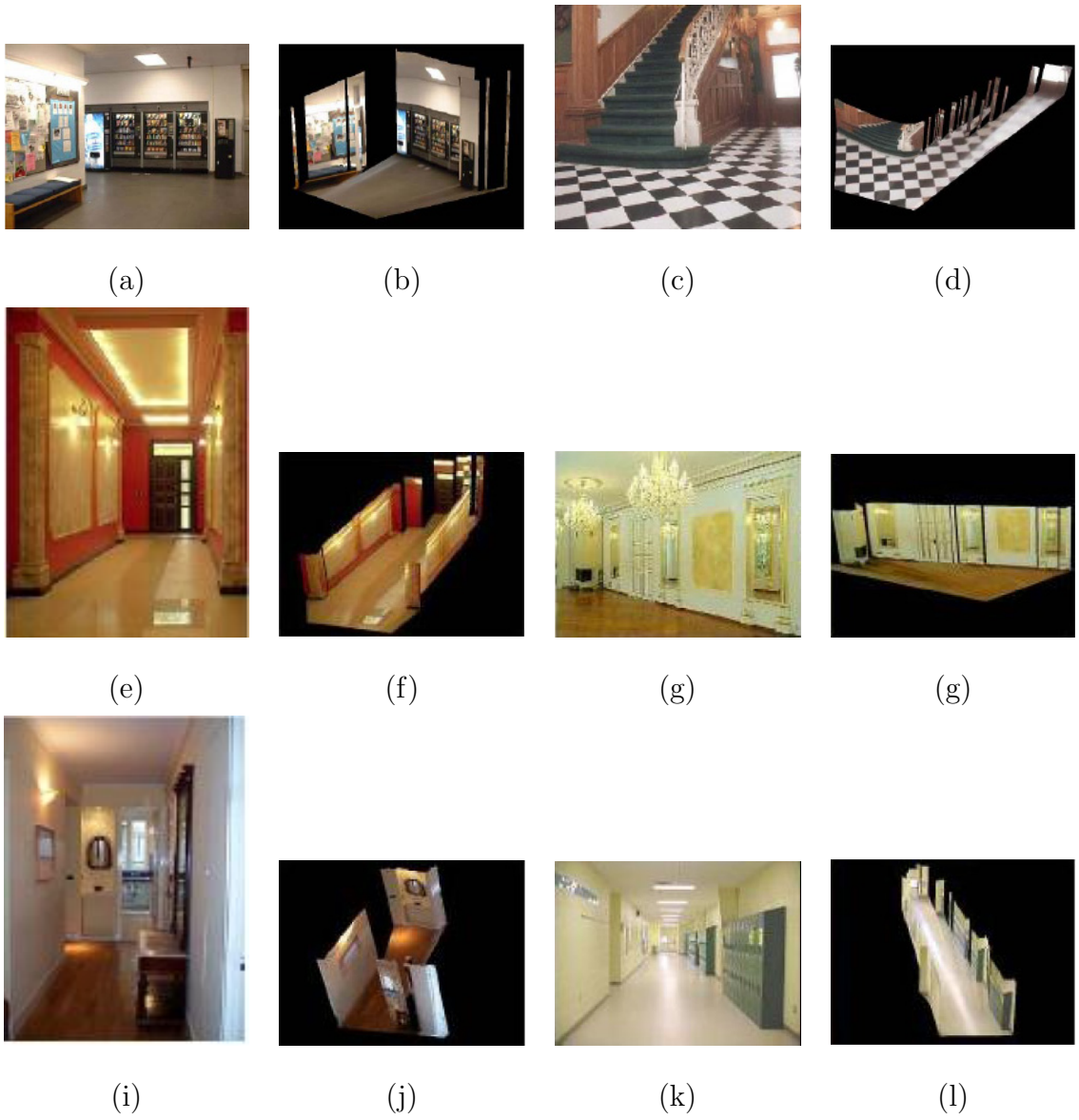


Figure 2.6: A dynamic Bayesian network (DBN) based 3D models. (a-b) represent original image and the constructed 3D model of the indoor scene using DBN. (c-d) represent original image and constructed 3D model of the indoor scene using DBN. (e-f) represent original image and the constructed 3D model of the indoor scene using DBN. (g-h) represent original image and the constructed 3D model of the indoor scene using DBN. (i-j) represent original image and the constructed 3D model of the indoor scene using DBN. (k-l) represent original image and the constructed 3D model of the indoor scene using DBN. The images are obtained with a calibrated camera from Stanford University. Reprinted from "Dyn06" (n.d.-b).

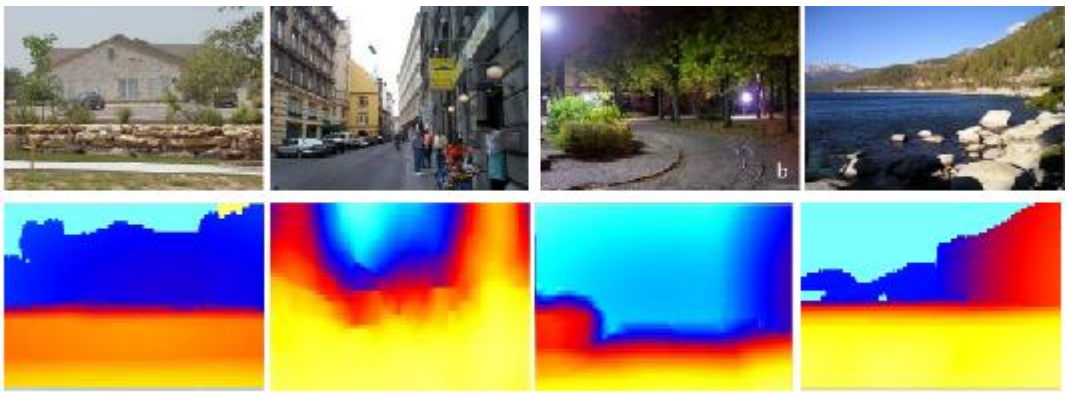
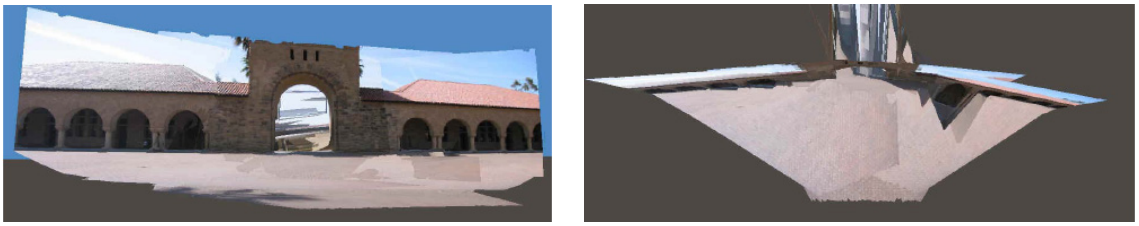


Figure 2.7: Images and their depth maps. Original images are in the top row and depth maps are in the bottom row. Colors represent depth. This is a log scale representation. Yellow is closed, which is followed by blue. Reprinted from Saxena et al. (2007).

(a) (b) (c) (d)



(e) (f)

Figure 2.8: Monocular based 3D models using Markov Random Field (MRF). Original images and their 3D models.(a, b, and c) represent original images. (d, e, and f) represent 3D models. Reprinted from Saxena et al. (2007).

Figure 2.9: Depth map of two images, using a stereo system. Reprinted from Saxena et al. (2008).

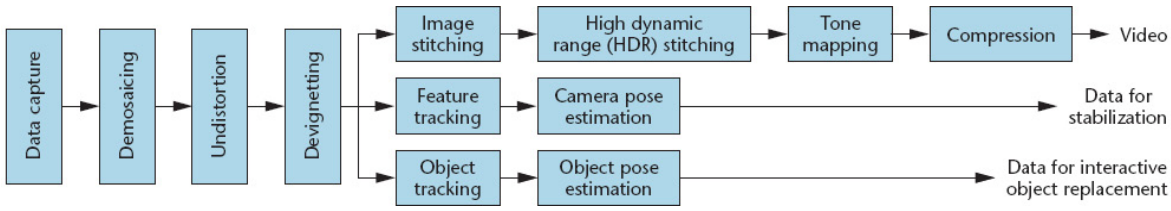


Figure 2.10: Rendering of pipelines and video-processing pipelines for image-based interactive tours. Reprinted from Uyttendaele et al. (2004).

2.6 Image Based Rendering

Uyttendaele et al. (2004) present an image-based rendering system (Figure 2.10) and claim it takes the held step closer to 3D reconstruction. They propose a first-ever system to build fully interactive, photo-realistic image based tours on a desktop computer at or above broadcast resolutions and frame rates. They avoid still photography and 3D reconstruction and as it tends to be brittle. Although it is true that image-based rendering has led to the largest scale environments by far 3D reconstruction is catching up. See for instance Pollefeys et al., 2004; "Iso09", n.d.-a; Saxena et al., 2007; "Dyn06", n.d.-b; and Saxena et al., 2008. In this works, still images have played a significant role.

2.7 Structured Lighting

Projection of light pattern at known angle on an object is termed as structured lighting. The light pattern can be a plane, grid, or any complicated shape. We can extract 3D information by scanning the object with light (Figure 2.11). A report written by Moravec (1996) demonstrates transformion of stereoscopic (images which are obtained from a camera implanted on a robot) into 3D maps of the robot's environment. The method achieves 100 to 1000 times improved maps.

2.8 MonoSLAM

Davison, Reid, Molton, and Stasse (2007) propose a dynamic algorithm that extracts the 3D trajectory of a single camera, moving rapidly in unknown scenes. They call their algorithm MonoSLAM, the first successful SLAM (Simultaneous Localization and Mapping) application to using a monocular camera in real time. Their main contribution is the creation of a sparse map of natural landmarks with a probabilistic approach. Their work extends the application of SLAM to real time video processing . A system such as MonoSLAM could potentially be used as the basis for a more photorealistic 3D reconstruction.

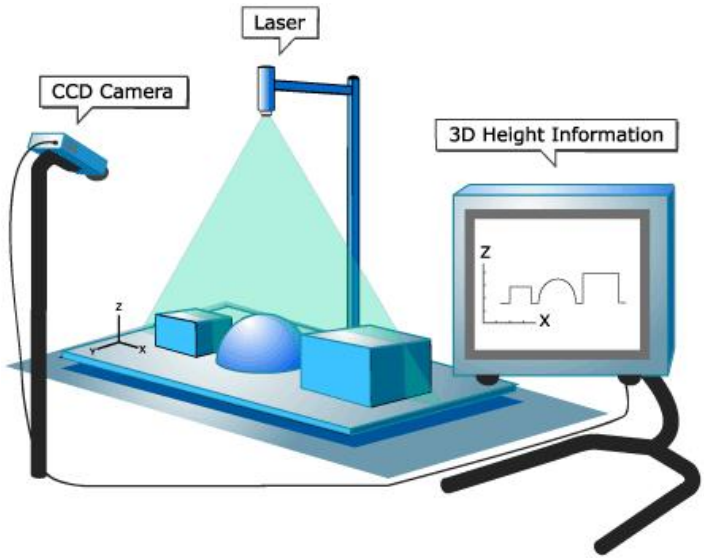
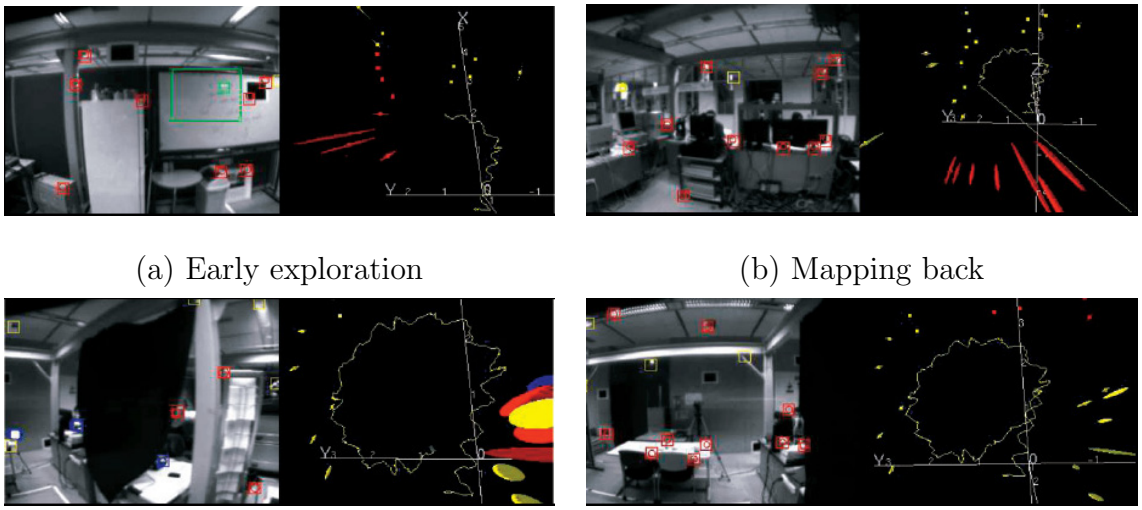


Figure 2.11: Structured Lighting. Extracting 3D information using structured lighting. Reprinted from *What is Structured Light?* (n.d.).



(a) Early exploration

(b) Mapping back

(c) Near the loop close

(d) Drift corrected

Figure 2.12: The images of humanoid robot walks in a circular loop of radius 0.75 m, using MonoSLAM. Reprinted from Davison et al. (2007).

2.9 Features detection

2.9.1 Scale Invariant Feature Transform (or SIFT)

Lowe (2004) has developed a method that extracts distinguishable features from images, which are useful for matching between different views of a scene (Figure 2.13). The drawn features are, not affected by rotation, translations or scale of images. SIFT also provides robust matching in cases of significant SIFT features distortion, variations in 3D view, noise, and variance in illumination. SIFT features have the potential to secure their identity when comparing them with a large database of features drawn from different images. These features can also be utilized for object recognition. The recognition procedure is based on matching features using a fast approximate nearest-neighbor algorithm, followed by a Hough transform, or other method identify clusters due to a single object. Verification is obtained through least squares methods. Using this approach, recognition can identify objects among clutter and occlusions in close to real time performance. According to Lowe (2004), some of SIFT's important applications are:

- 3D reconstruction.
- Tracking and segmentation.
- Robot localization, in which the main task is to maintain a maximum number of of identified landmarks during motion.
- Assembly of image panorama.
- Epipolar calibration.
- Any other application that requires identification of matching locations between images.

2.9.2 SURF (Speeded Up Robust Features)

Bay, Ess, Tuytelaars, and Gool (2008) present SURF, which is a scale invariant, rotation invariant detector and descriptor. It is faster than previous methods like SIFT (Scale Invariant Feature Transform). SURF uses integral images for convolution builds on the strong points of existing detectors and descriptors and while simplifying those methods. This leads to detection, description and matching steps. Some of the main applications of SURF are:

- Camera calibration
- 3D reconstruction
- Registration, and object recognition

The results suggest that the performance of their Hessian approximation is equivalent or is better than existing best-point detectors. The frequent repetition is favorable for camera calibration. The precision of point detection has a direct consequence on the reliability

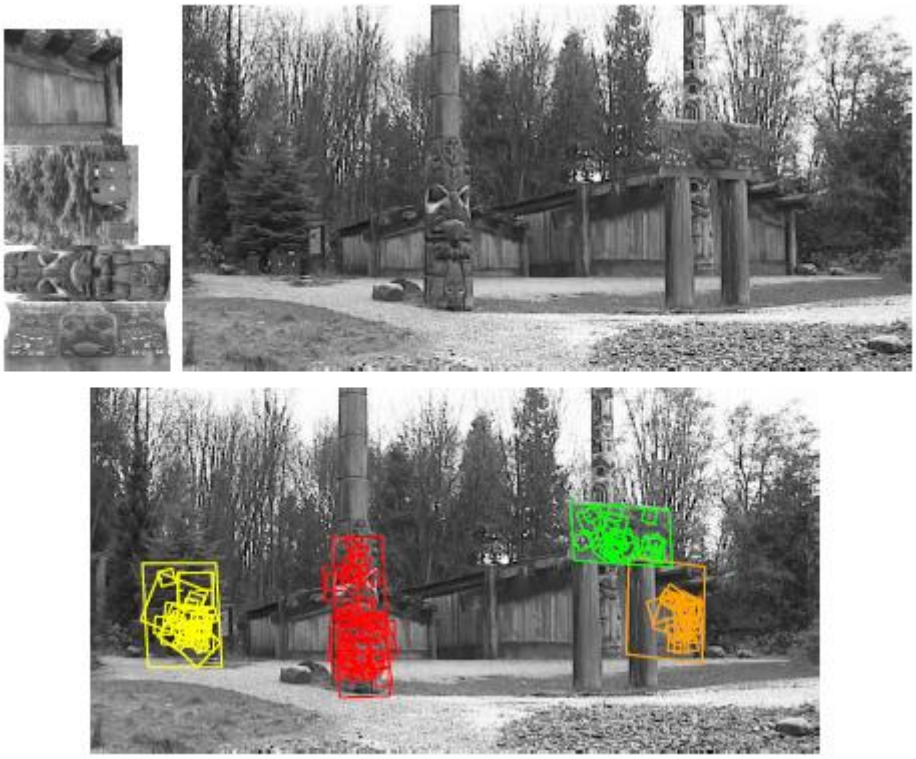


Figure 2.13: SIFT . The upper left shows the training images and the upper right shows a test image with different views of the objects in the the upper left images. The lower image shows the recognized regions. Reprinted from Lowe (2004).

of results for self calibration. This results have impact on the quality of the resulting-3D model (Figure 2.14). The significant result is the speed of the detector without any direct intervention with any optimization algorithm. A Laplacian-based indexing technique makes the matching faster without any degradation of the performance which is the basis for many online computer vision applications. The results recommend SURF for object detection, object recognition and image retrieval.

Figure 2.14: SURF's application. The image on the left shows one of the 13 images used for camera calibration and image on the right shows the sparse 3D model along with the position of the reconstructed cameras. Reprinted from Bay et al. (2008).

2.10 Scaling of Maps

In the opinion of Konolige (2004) scientists are nowadays attracted toward the scaling of their maps after successful small-scale indoor and outdoor mapping. Large maps are difficult to construct as the complexity to build them is typically $O(n \log n)$. Konolige presents a general method for controlling the growth in computation for building a map. Map creation from a mobile robot becomes a computational problem as the map scale gets larger. There are two types of metric mapping methods: Features (or landmarks) and dense surface information, that do not discriminate between features. The author emphasis on the map as a set of robot poses with constraint among them. This kind of representation scales well, when the area of map grows, as it only presents local restrictions among poses. They give an $O(n)$ algorithm for simple loop closure, n represent the number of poses along the loop. Using sparse matrix structure of Consistent Pose Estimation (CPE) for multiple loop they develop an efficient $O(n \log n)$ algorithm. Now after this research, it is expected that map region comprising of 10^5-10^6 poses and cover a million square meters will be mapped.Figure 2.15 represents a map of 1900 nodes. The map is created by putting seeds arbitrarily in an area and growing them until they touch their borders (Figure 2.15).

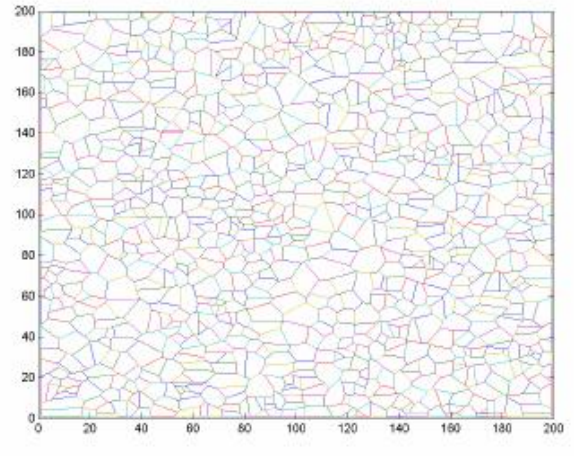


Figure 2.15: Multi-loop Map. An abstract map glue together by pose using 1900 nodes. Reprinted from Konolige (2004).

2.11 2D Maps

Elfes (1987) says a sonar system is ideally suited for an autonomous mobile robot performing duties in unknown terrain. A multilevel description of the robot's environment is created using sonar range data. The sonar readings are integrated using probability profiles to distinguish empty and occupied areas. Usually, the real time information captured with the help of sensors is quite sufficient. Few of the range based systems make denser decryptions but are unable to cope with measurement errors. In Figure 2.16 object extraction is shown. The map is obtained by scanning the sonar map and region coloring scheme is used to identify the objects. Objects are polygonal-shape and numbers are used to identify them.

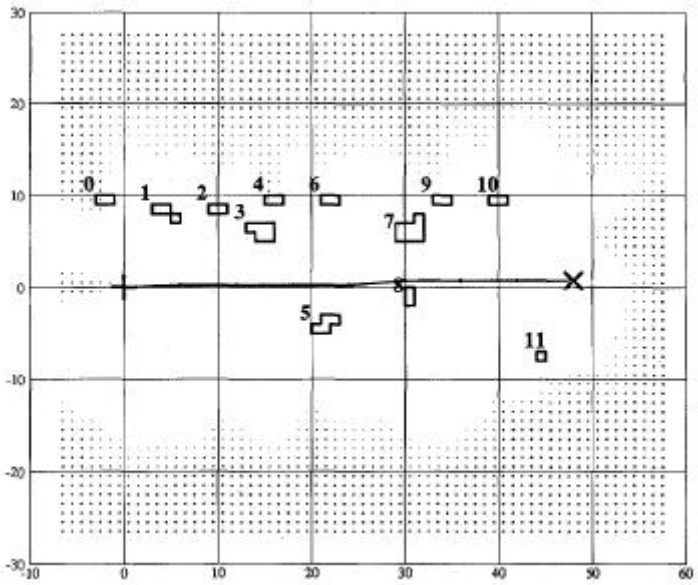


Figure 2.16: Sonar map used to identify objects. Objects are polygonal in shape and their numbers' are used to identify them. Reprinted from Elfes (1987).

Thrun, Burgard, and Fox (2000) present an algorithm for concurrent mapping and localization. This algorithm is incremental in nature and mostly used for mobile robots with 2D laser range finders. They apply a fast scan matching paired probabilistic method, which is used for localization. A multi resolution approach is employed from the computer graphics field of area to generate compact 3D maps and the data is input using a dual laser system. They build robust 3D maps of a large cyclic environment in real-time and without the presence of any odometric data. Most of the incremental methods suffer heavily when mapping large cyclic environments. The robot's cumulative error can grow in case of mapping large cyclic environments. The remedy is to close the cycle for that error but needs to be corrected backwards in time, which most of the current systems are incapable. They combines ideas from the EM method, and their method is also incremental. The idea is to combine key element of the EM based approach i.e. posterior estimation with that of incremental map construction using maximum likelihood estimators, which is the key of previous incremental approach. The posterior estimation to gather data accumulate from more than one robot. As a consequence, robots globally localize themselves in a map, which is constructed by any other robots. Finally, they build less complex 3D models of indoor scenes, which are based on low resolution algorithm. The results confirm that they build 3D maps with two laser range finder implanted on a robot with accuracy and filter out outliers based on distance. They also build 3D maps without odometry information with the limitation that the environment

2.12 Delaunay Triangulation

In Figure 2.17 (a) represents a terrain, (b) Delaunay triangulation, and (c) Polyhedral terrain. In this study I approximate a scene, by building a polyhedral model from a set of sample points. The Delaunay triangulation maximizes the minimum angle of each triangle. It can be obtain from the Voronoi diagram, by triangulating faces with more than three vertices, we can obtain Delaunay triangulation. According to Berg, Cheong, Kreveld, and Overmars (2008) there is another way to obtain the Delaunay triangulation, using a randomized incremental approach (Figure 2.19).

2.13 Mesh Generation

Lorensen and Cline (1987) have propose a classic algorithm for mesh generation named marching cubes. This algorithm generates triangle models of uniform density based on 3D density data. The triangle topology is based on a divide and conquer algorithm. It is used to generate inter-slice connectivity. A case table is enumerates triangle topology. The algorithm is used in scan-line sequence and compute triangle vertices using linear interpolation. The authors find the slope of the actual data, normalize it, and utilize it as a basis for shading their models. The different results from computed tomography (CT), magnetic resonance (MR), and single-photon emission computed tomography (SPECT) confirm the standard and functionality of marching cubes. The algorithm utilizes the use case table of edge intersection. It tells, how a surface cuts through each in a 3D data set. Using surface cutting and connectivity can reduce large number of triangles in the models. Figure 2.20 shows marching cubes algorithm identify the surface in a logical cube build from eight pixels. From two adjacent slices, it takes four for each pixels.

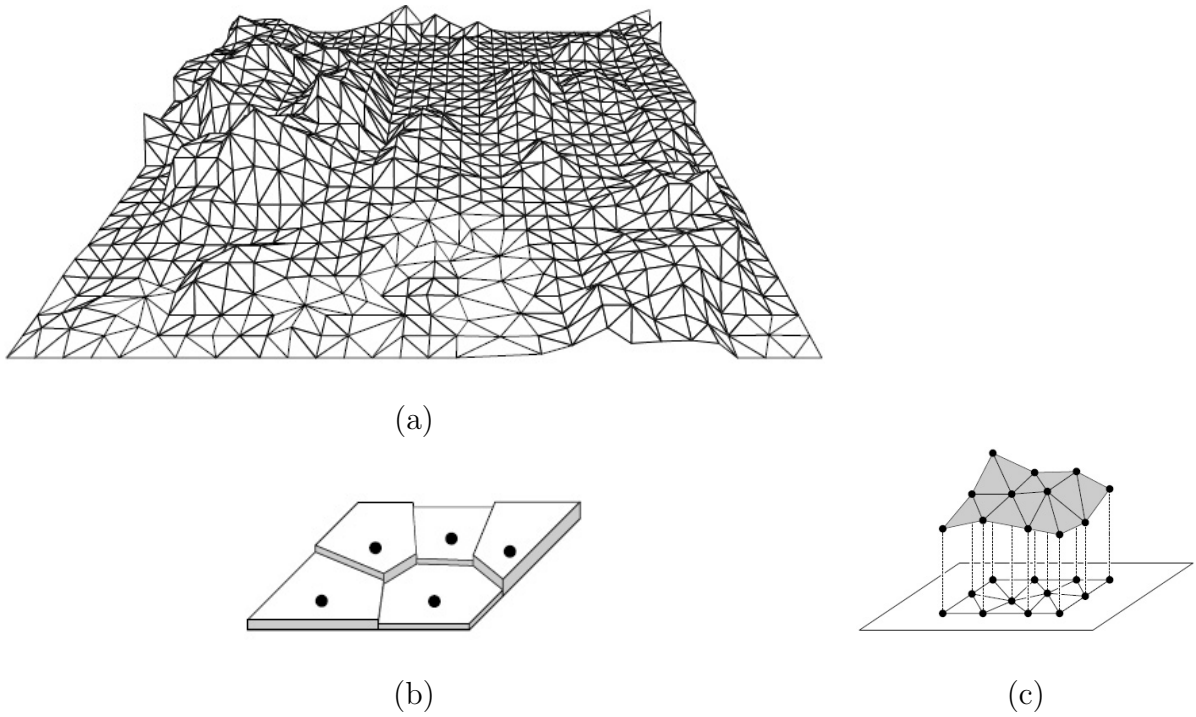


Figure 2.17: Delaunay triangulation. (a) A view of terrian (b) Delaunay triangulations and (c) Polyhedral terrain. Reprinted from Berg et al. (2008).

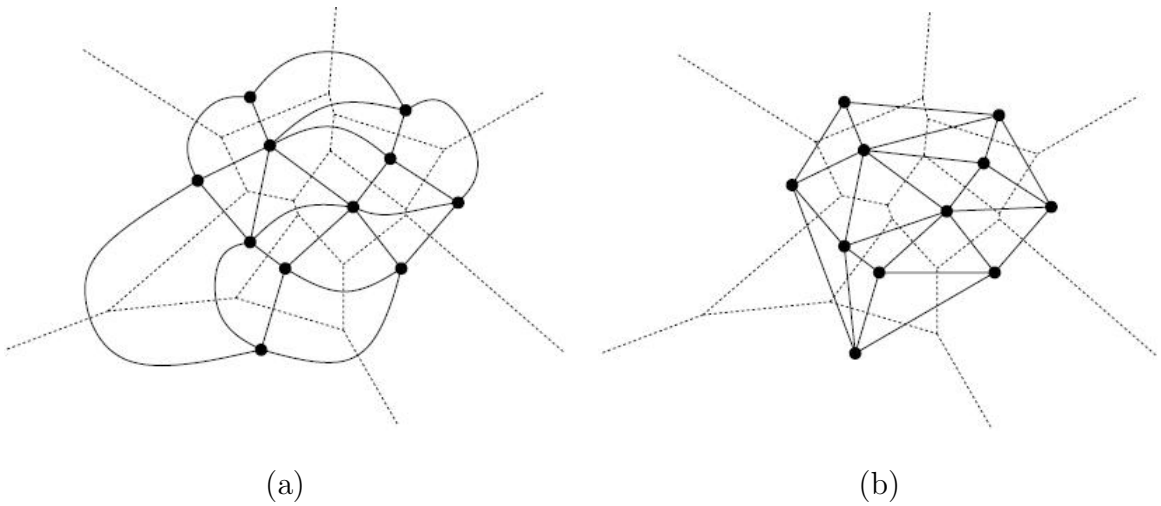


Figure 2.18: A randomized incremental approach for getting a Delaunay graph. A Delaunay diagram from a Voronoi diagram (a) A Voronoi diagram (b) The Delaunay graph from the Voronoi diagram. This is the first approach of getting a Delaunay graph. Reprinted from Berg et al. (2008).

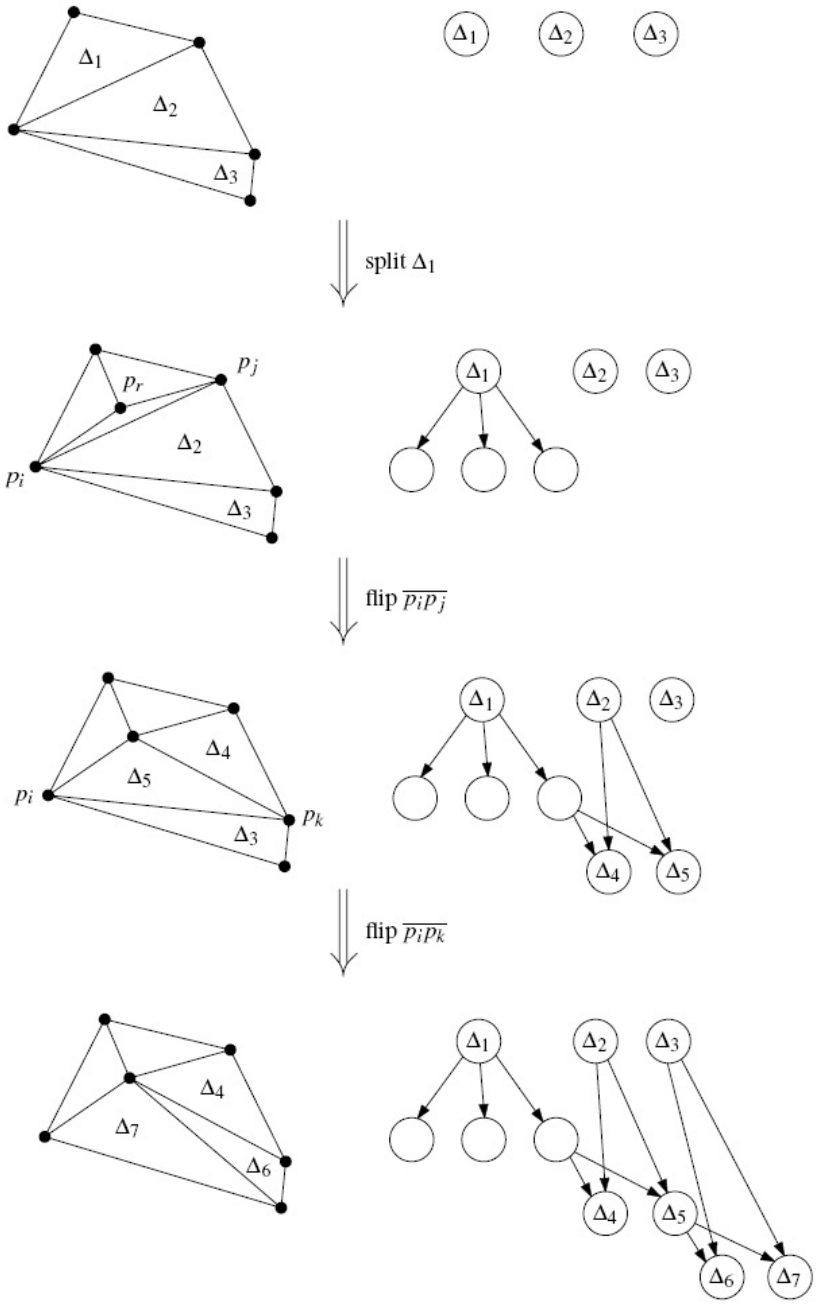


Figure 2.19: This is the second approach of obtaining a Delaunay diagram, which is called the randomized incremental approach. The consequences of inserting point p_r into triangle $\Delta 1$ on the data structure D. If D is not change, then that part of D is not included in this Figure. Reprinted from Berg et al. (2008).

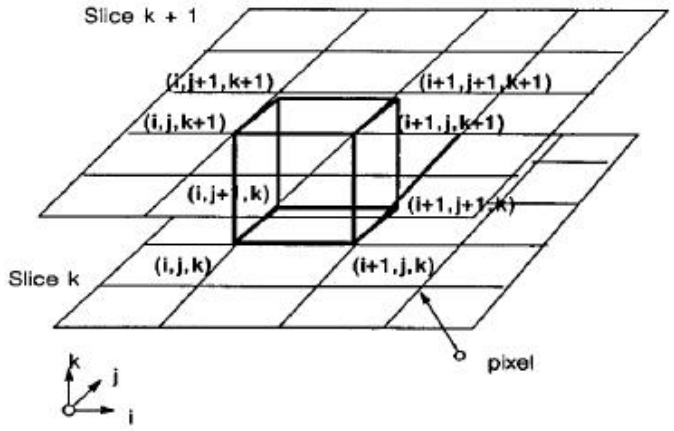
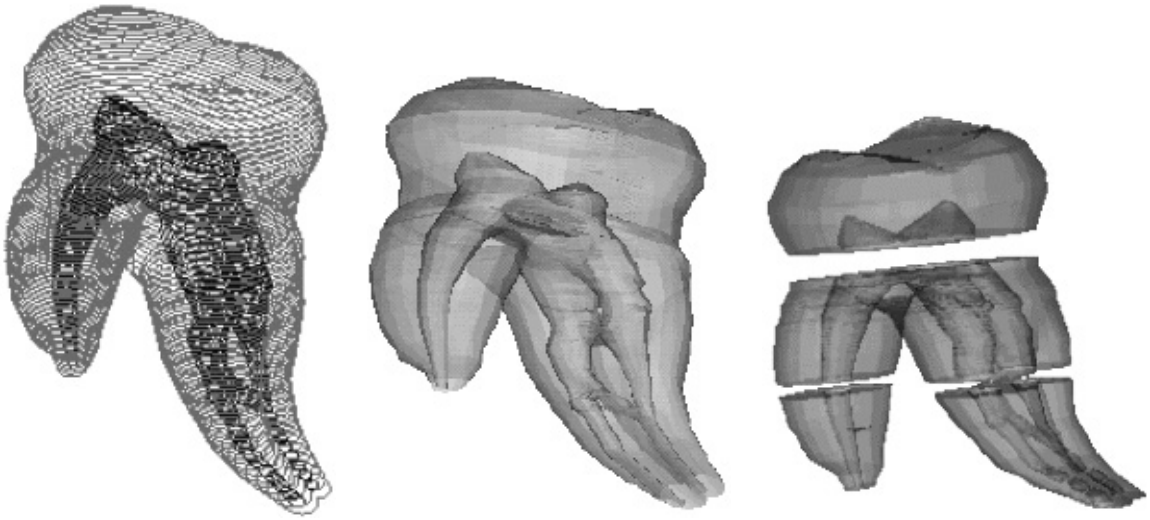


Figure 2.20: Marching cubes identifies the surface in a logical cube build from eight pixels. From two adjacent slices, it takes four for each pixels. Reprinted from Lorensen and Cline (1987).

Nonato, Minghim, Oliveira, and Tavares (2002) propose a new method for volumetric reconstruction of 3D objects from planar surfaces, based on Delaunay triangulation. It is helpful in 3D model reconstruction from the prospective of generating a mesh that is feasible for interaction and simulation assignments. Figure 2.21, shows the external surface of the tooth (a) and its reconstruction using two approaches (b-c).



(a) (b) (c)

Figure 2.21: Tooth. (a) Set of contours. (b) Nonato's algorithm for reconstruction. (c)Using Geigers algorithm for reconstruction. Reprinted from Nonato et al. (2002).

Cuadros-Vargas, Lizier, Minghim, and Nonato (2008) redesign a past method for mesh generation based on intensity images called Imesh. This reformulation has a very good effect on Imesh and improves it all around, now this algorithm is capable of generating meshes from different kinds of images and it can be used for multiple purpose. Figure **??** shows object and its sub meshes in iterations.

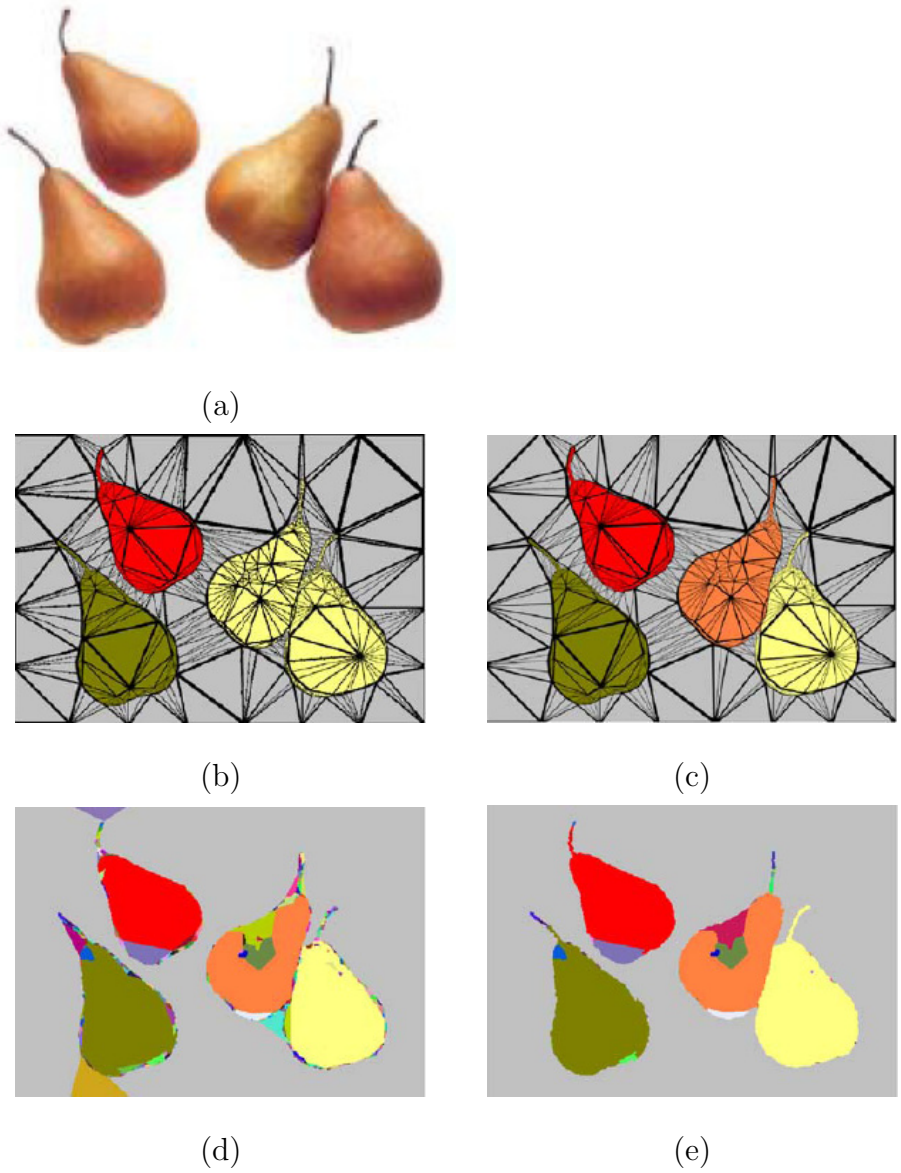


Figure 2.22: Mesh generation. (a) Original image.(b) Segmented meshes by pixels. (c) Same by pixels and geometrical properties. (d) Sub meshes (first iteration). (e) Sub meshes (second iteration). Reprinted from Cuadros–Vargas, Lizier, Minghim, and Nonato (2008).

Chapter 3

Methodology

This chapter is devoted to the explanation of my research methodology, which is based on my objectives. The primary goal of this research is to design a textured 3D model based on sparse points, using Delaunay triangulation. This chapter is composed of two sections: system overview and system design and implementation.

3.1 System Overview

This section describes the main flow and functionality of the system. We can get an overview of calibrated reconstruction by referring to Figure 3.1. My research is based on sparse 3D points, using a monocular camera. For that, I need a sequence of images from a single camera, which will be calibrated. The difference between calibrated and uncalibrated reconstruction is also visible in Figure 3.1. I use the data set (of images) of (Zask, 2009), due to a shortage of time. I find the key points in the pair of images. The new image's key points are compared with the previous image's key points to find correspondences between two images. The points' correspondences in two images are used to estimate the camera pose, triangulation is used to reconstruct the 3D points. The reprojection errors are minimized using Sparse Bundle Adjustment (Lourakis & Argyros, 2004). Then the images are rectified. 3D surface model is built incrementally and to visualize it, a textured map is used.

3.2 System Design and Implementation

This section of my research, is dedicated to the explanation of Figure 3.1. I also discuss some implementation details in this section, for instance: SIFT (Scale Invariant and Feature Transform), RANSAC (Random Sample and Consensus), BA (Bundle Adjustment), and Delaunay triangulation.

3.2.1 Data set

I take the data set of Ran ("Iso09", n.d.-a). There are four types of scenes in their data set: (a) long corridor, (b) 360 degrees of a small office, (c) L-shaped corridor, and (d) T-shaped corridor. I am interested in the L-shaped corridor.

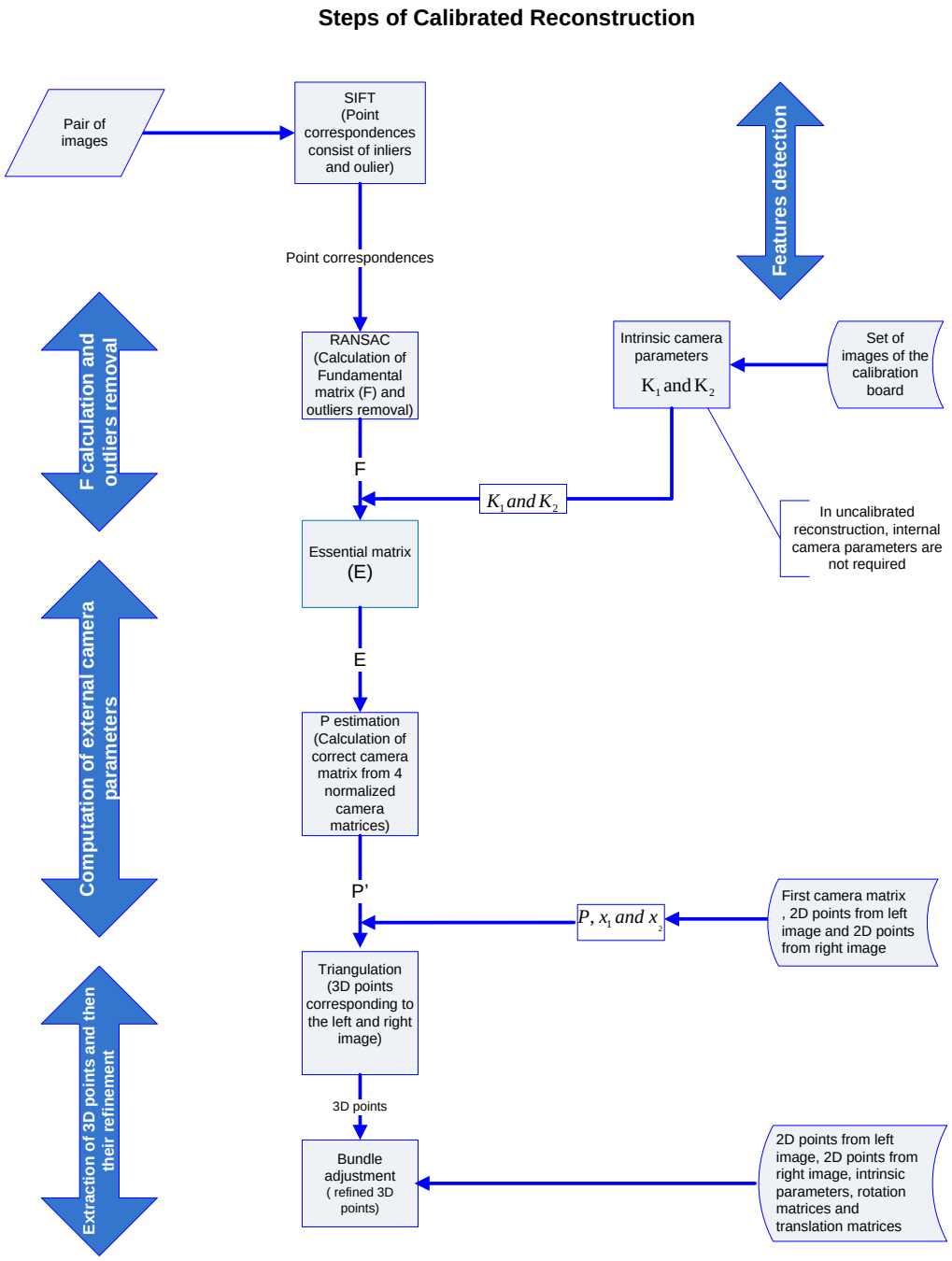


Figure 3.1: Main steps for calibrated reconstruction.

3.2.2 Camera Calibration

Camera calibration is the procedure of finding the matrix P, the camera projection matrix, from 2D-3D point. It can be regarded as a procedure to separately find intrinsic parameters K and extrinsic parameters T. One important limitation of camera calibration is that we obtain those images and from them the set of features (2D points), whose real 3D points are known a priori. To calibrate the camera, most researchers use Camera Calibration Tool Box for Matlab Bouguet (2004). They take several pictures of calibration board from different angles, in order to check the accuracy of the estimates. The parameters to be estimated are the intrinsic parameters (focal length, principal point, skew coefficient and distortions), and the extrinsic parameters (rotations and translations). According to Heikkila and Silven (1997), when we use real cameras, the images always contain noise. It is always good to assume that this noise is white Gaussian, for the systematic part of measurement error. It is essential to undistort the images for better 3D reconstruction. For distortion removal, the same Matlab Tool Box for camera calibration (Bouguet, 2004) is used.

3.2.3 Feature Detection in Images using SIFT

A feature point should be differentiated from its neighboring image points (Pollefeys et al., 2004). SIFT selects key points that can distinguished from their neighboring image points. The extracted features should not to be affected by rotation and translation of image (Lowe, 2004). SIFT is partially invariant to changes in illumination and 3D viewpoint. Even when there is no initial guess about the camera motion, SIFT is recommended. These are the reasons, why I select SIFT for key point detection from images. It is better than cross correlation, which is affected transformation. Cross correlation also needs an initial guess of camera motion.

3.2.4 Correspondence Detection and Calculation of Fundamental Matrix using RANSAC

RANSAC extracts inliers and removes outliers, which are the primary weakness of least squares methods. Least squares methods tr use both inliers and outliers for the best fit of a model. RANSAC's algorithm is explained in Figure 3.2. I use RANSAC to estimate the fundamental matrix (F) and to find a set of interest point in correspondences. (Hartley & Zisserman, 2004). For this purpose, I follow the algorithm 11.4 of Hartley and Zisserman (2004).

3.2.5 Computation of the Essential Matrix

The essential matrix (E) can be computed from the fundamental matrix (F) and the camera calibration matrix (K) using the following relation:

$$E = K^{'} F K. \qquad \text{(Equation 3.1)}$$

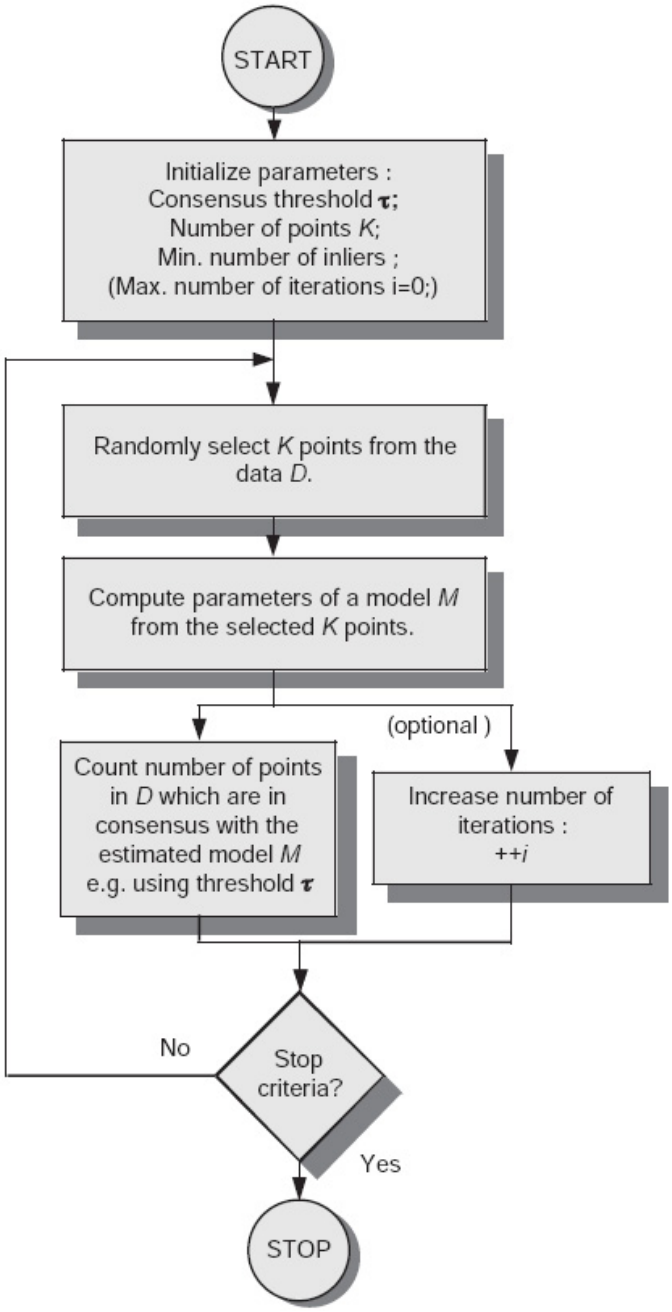


Figure 3.2: RANSAC algorithm. Reprinted from Cyganek and Siebert (2009).

We need the internal camera calibration matrices K and $K^{'}$ of the two cameras. Since same camera is used for the sequence of images and even the zoom of the camera was not change, $K^{'} = K$.

3.2.6 P Estimation

We calculate E using Equation Equation 3.1. When I get E, I use it to get the correct camera matrix. We assume that the first camera matrix is

$$P = [I|0] \qquad \text{(Equation 3.2)}$$

According to result 9.19 of Hartley and Zisserman (2004), if P and E are given, then there are four possible choices for the second camera matrix, namely $P^{'}$:

$$P^{'} = [UW^{T}V^{T}|u_3] \qquad \text{(Equation 3.3)}$$

$$P^{'} = [UW^{T}V^{T}| - u_3] \qquad \text{(Equation 3.4)}$$

$$P^{'} = [UW^{T}V^{T}|u_3] \qquad \text{(Equation 3.5)}$$

$$P^{'} = [UW^{T}V^{T}| - u_3] \qquad \text{(Equation 3.6)}$$

I take SVD $U.\text{diag}(1,1,0).V^{T}$ of normalized E, where

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In order to decide about the correct camera matrix, it will be sufficient to check a single point to decide, whether it is in front of both the cameras. In other words, I will follow the following procedure:

- Take a test point pair from my set 2D of point correspondences.
- Use x_1 (a point in image1), P, x_2 (a corresponding point in image2), one of the camera matrices $P^{'}$ (from the four camera matrices above), and then I perform triangulation.

- Check, which of the above camera matrices $P^{'}$ will give me positive depth.
- Select the $P^{'}$, which gives a positive depth in both cameras.

3.2.7 Triangulation and 3D Points

I use P and $P^{'}$, x_1 (a set of 2D points from image1), x_2 (a set of corresponding 2D points from image2). Then I use triangulation to obtain the 3D points.

3.2.8 Minimization of Reprojection Error Using BA

After I get 3D points, my next important task is reduce the reprojection error defined by the equation given below:

$$\sum_i d(x_i, X_i P)^2 \qquad \text{(Equation 3.7)}$$

I use Sparse Bundle Adjustment (SBA) by Lourakis and Argyros (2004) for this task. There are some limitations of BA (Hartley and Zisserman (2004)):

- Good initialization is a must for BA.
- Since there are a large number of parameters involved, so it can become huge minimization problem.

In my experiments I find that estimation of $P's$ using E and triangulation to obtain X give good enough initial results, and for two images, the SBA package is more than sufficient.

3.2.9 3D Modeling using OpenGL

This is the last step for this research. The reconstructed model is rendered with texture mapping or it will be render without it. I use:

- OpenGL.

Chapter 4

Experiments and Results

4.1 Overview

This chapter describes the results of this research study. The main flow of my work is based on Figure 3.1. This chapter is divided in the following sections: data collection, camera calibration using calibration board, distortion removal from the set of images, bundle adjustment, Delaunay triangulation and OpenGL modeling.

4.1.1 Data collection

I utilized the data set from "Iso09" (n.d.-a). I utilized the data set from "Iso09" (n.d.-a).

4.1.2 Camera calibration using calibration board and distortion removal

I used the Camera Calibration Toolbox for MATLAB by Bouguet (2004). I took several pictures of calibration board with different angles, in order to calibrate the camera properly. In step1, I got the calibrated images from the set of images, using the toolbox. In step2, I got the boundary of the calibration board. In step3, I got the guessed grid corners. In step4, I got almost exact extracted corners of the grid. This is visualized in Figure 4.2. In the second phase, I got the reprojection error of the images using the toolbox. Then, I got the visualization of the extrinsic parameters (both world centered and camera centered) with respect to the calibration board. I got the image points and reprojected grid points on the same graph. This is visualized in Figure 4.3. Finally, the radial distortion from the images was removed in advance, in order to get better results from the images.

4.1.3 Results of Bundle adjustment:

- Starting BA with fixed intrinsic parameters.
- SBA using 733 3D pts, 4 frames and 1466 image projections, 2223 variables.
- Method SBA-MOTSTRUCT, expert driver, analytic Jacobian, fixed intrinsics, without covariances.
- SBA returned 150 in 150 iter, reason 3, error 0.0717044 [initial 1.0647e+10], 177/150 func/fjac evals, 176 lin. systems.
- Elapsed time: 0.77 seconds, 770.00 msecs.

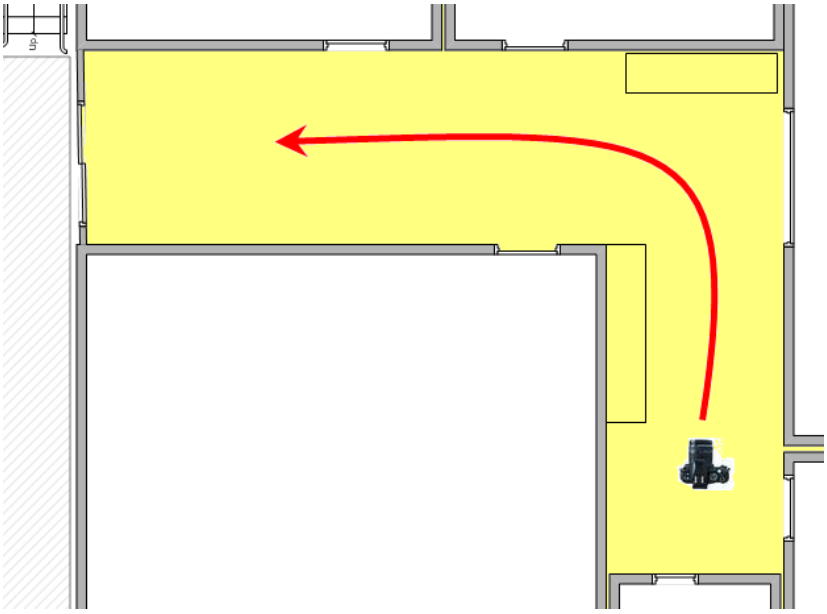


Figure 4.1: Ran's data set. I select L-shaped for my research study. Reprinted from Zask (2009).

4.1.4 Procedure for 3D modeling

- First I obtain the 3D points.
- Then I find the minimum and maximum from the x, y and z coordinates of these points.
- I decided about the granularity of the grid. Granularity, is the extent to which a larger entity is subdivided.
- I map all the 3D points to the nearest grid cell center.
- I increased the count of obtained 3D points for this grid cell center.
- I construct a new 3D point set consisting of the grid cell centers with a count of 3.
- Then I find the projections of these 3D points into the image.
- I get the new sparser set of 2D points.
- I get the 2D Delaunay triangulations of these points.
- I create a 3D model consisting of the triangles from the previous step, using the 3D positions rather than the 2D projections.

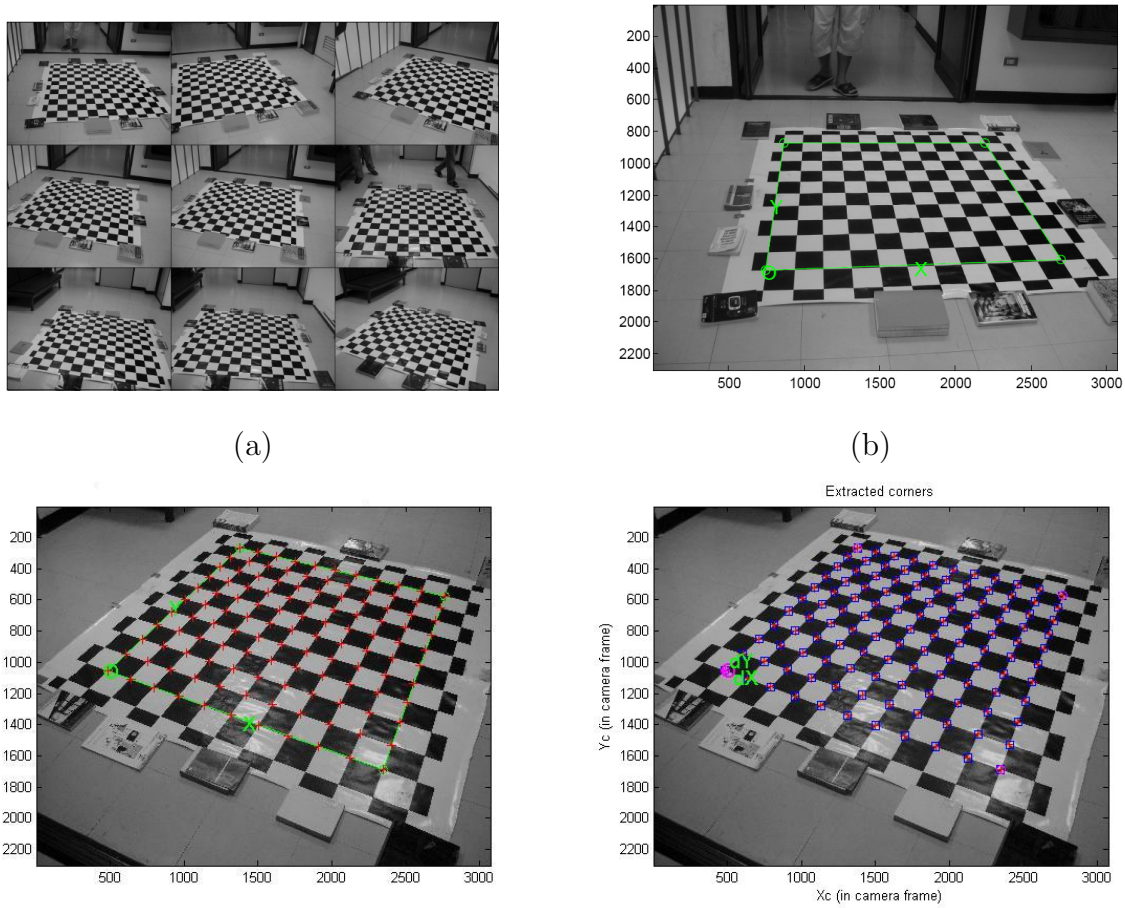


Figure 4.2: Camera calibration. (a) Calibrated images. (b) Boundary of calibration grid. (c) Red boxes represent guessed grid corners. (d) Extracted corners.

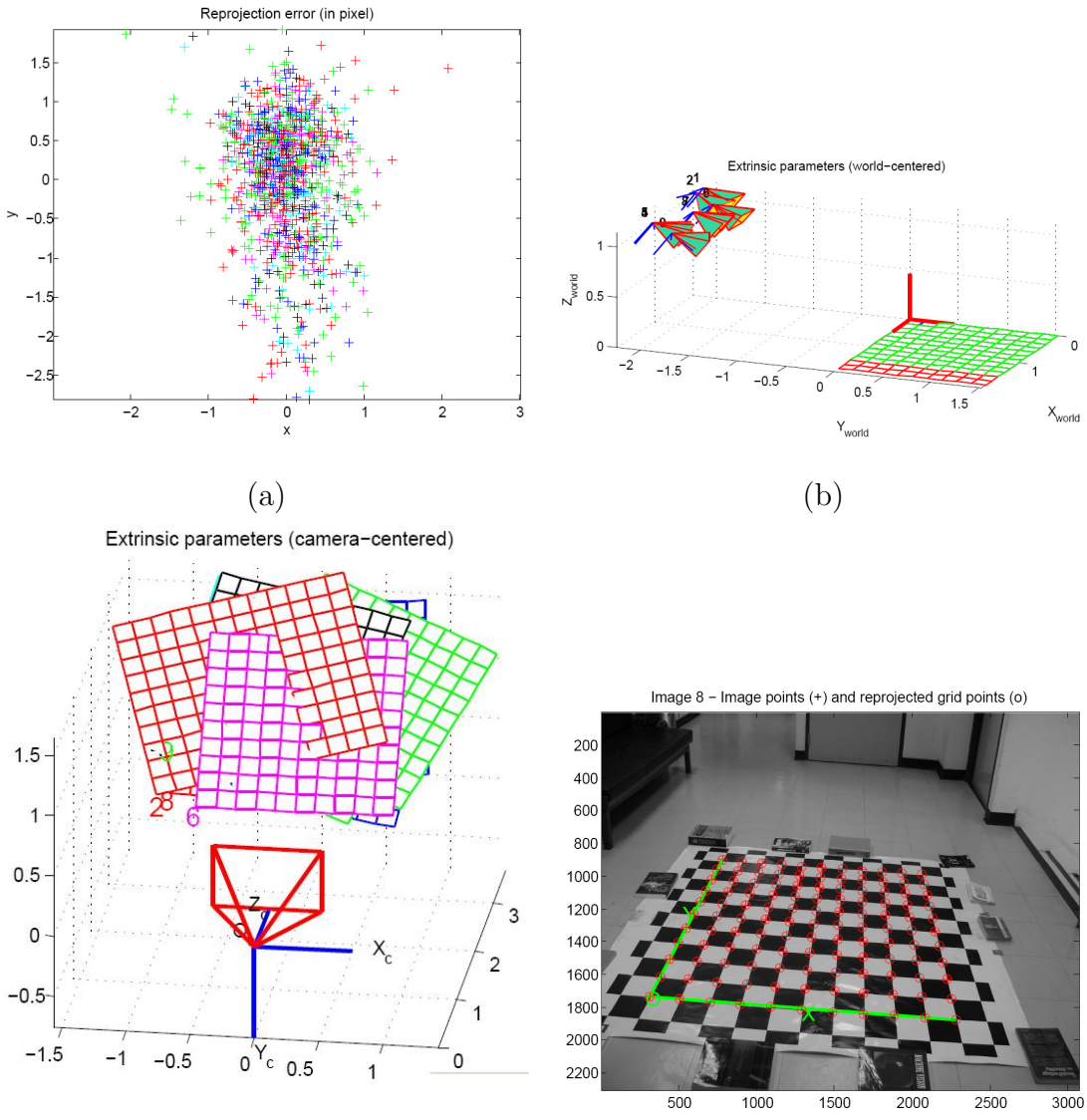


Figure 4.3: Camera calibration results. (a) Reprojection error plot. (b) The green pyramid represented different cameras' positions and orientations. (c) The camera reference frame was denoted by (O_c, X_c, Y_c, Z_c). The effective field of the camera was in the range of red pyramid. The effective field of view was defined by the extrinsic parameters (camera-centered) (d) Image points are denoted by (+) and reprojected grid points are denoted by (0).

Main Calibration step:

Initialization of the intrinsic parameters - Number of images: 9

Calibration parameters after initialization:

Focal Length: fc = [3210.51286 3210.51286]

Principal point: cc = [1535.50000 1151.50000]

Skew: alpha_c = [0.00000] => angle of pixel = 90.00000 degrees

Distortion: kc = [0.00000 0.00000 0.00000 0.00000 0.00000]

Main calibration optimization procedure - Number of images: 9

Gradient descent iterations: 1...2...3...4...5...Warning: it appears that the principal point cannot be estimated. Setting center_optim = 0

6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...done

Estimation of uncertainties...done

Calibration results after optimization (with uncertainties):

Focal Length: fc = [2778.38433 3516.50410] ± [64.51970 195.84811]

Principal point: cc = [1555.30301 2354.04520] ± [0.00000 0.00000]

Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees

Distortion: kc = [-0.07950 -0.02072 -0.03248 0.00002 0.00000] ± [0.00653 0.01132 0.00082 0.00102 0.00000]

Pixel error: err = [0.43269 0.82220]

Note: The numerical errors are approximately three times the standard deviations (for reference).

Main calibration optimization procedure - Number of images: 9

Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...done

Estimation of uncertainties...done

Calibration results after optimization (with uncertainties):

Focal Length: fc = [2775.94000 3510.05501] ± [68.39983 207.16821]

Principal point: cc = [1555.30301 2354.04520] ± [0.00000 0.00000]

Skew: alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees

Distortion: kc = [-0.07994 -0.02244 -0.03285 -0.00002 0.00000] ± [0.00739 0.01432 0.00093 0.00110 0.00000]

Pixel error: err = [0.41724 0.81688]

Note: The numerical errors are approximately three times the standard deviations (for reference).

Figure 4.4: The above numerical results show my attempt at calibration.

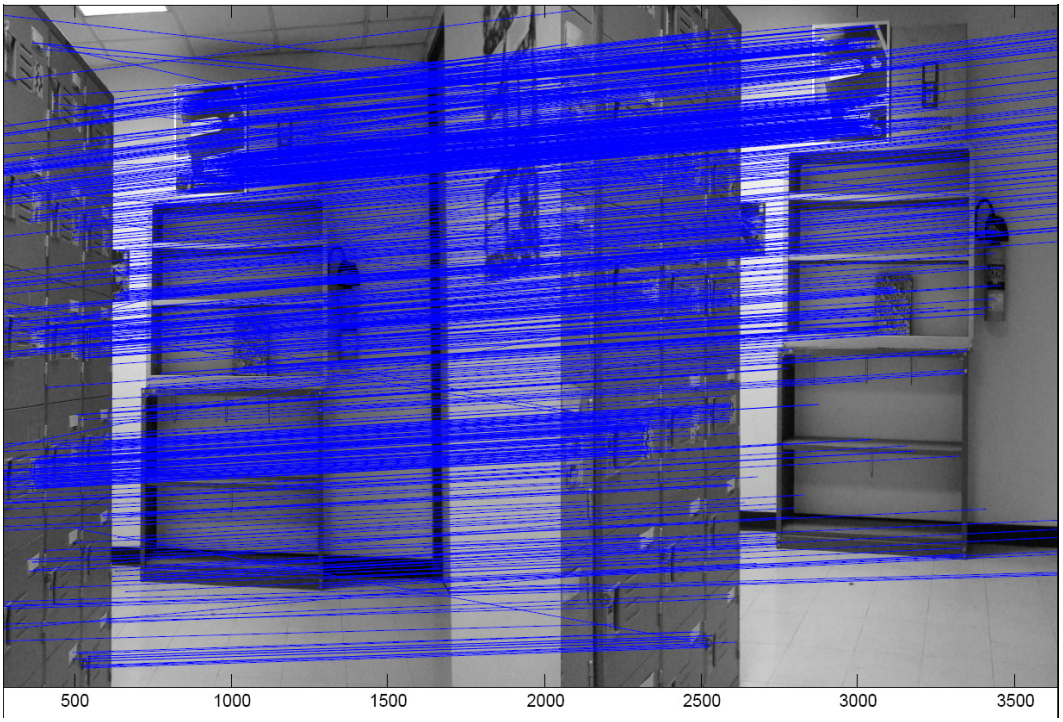
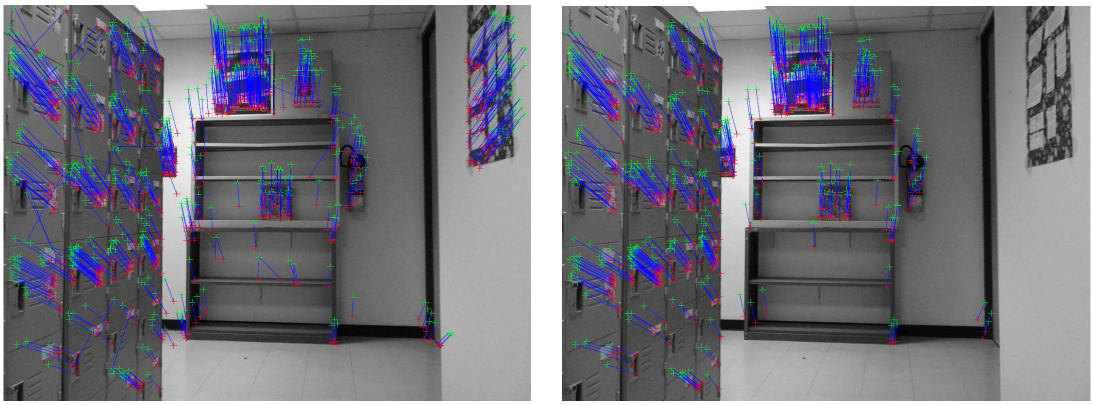


Figure 4.5: SIFT point matching. The left and right images were of size 2048x1536. SIFT found 5315 key points in 26 seconds from the left image and 5329 key points in 25 seconds from the right image. Also it found 857 matches in 37 seconds from both the images. I selected the distance ratio of 0.60.



(a) (b)

Figure 4.6: SIFT point matching after F estimation and outliers elimination. (a) Before removing outliers (b) After removing outliers using RANSAC.

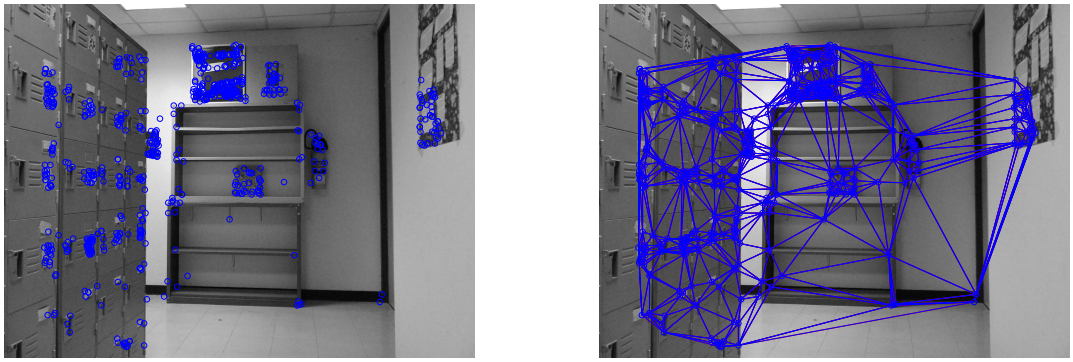
(a)

(b)

(c)

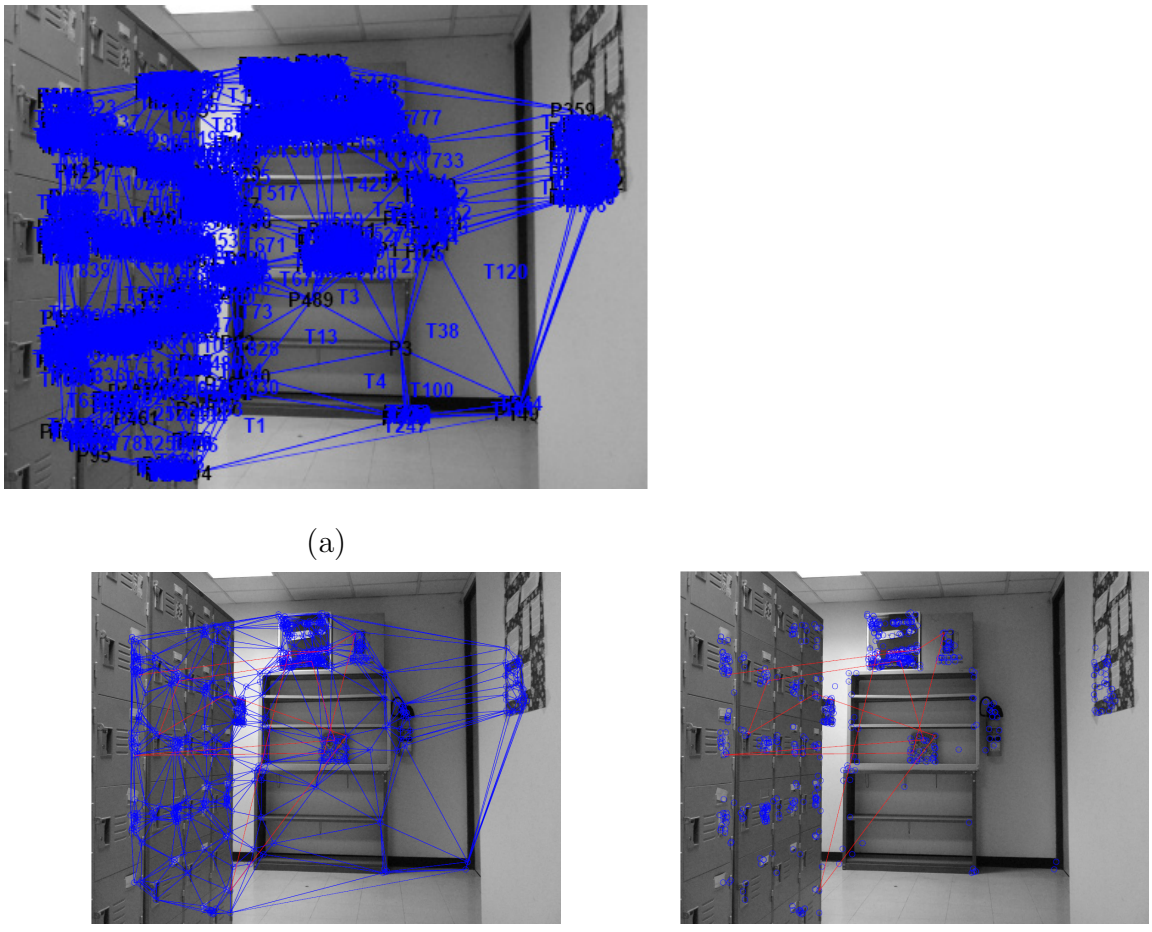
(d)

Figure 4.7: (a) Randomly selected few points, from left image. (b) Points on left image was represented by lines on the right (c) Randomly selected few points, from left image. (d) Points on left image was represented by lines on the right.



(a) (b)

Figure 4.8: (a) 2D points extracted from the image on LHS, using SIFT and then applying RANSAC to remove the outliers. (b) Delaunay triangulation corresponding to the 2D points extracted from the image on LHS.



(a)

(b)

(c)

Figure 4.9: (a) The Figure showed the triangles' number and their vertices.(b) The small circle (o) showed the 2D points, the blue triangles represented Delaunay triangulation and the red lines showed the convex hull. (c) 2D points and the corresponding convex hull.

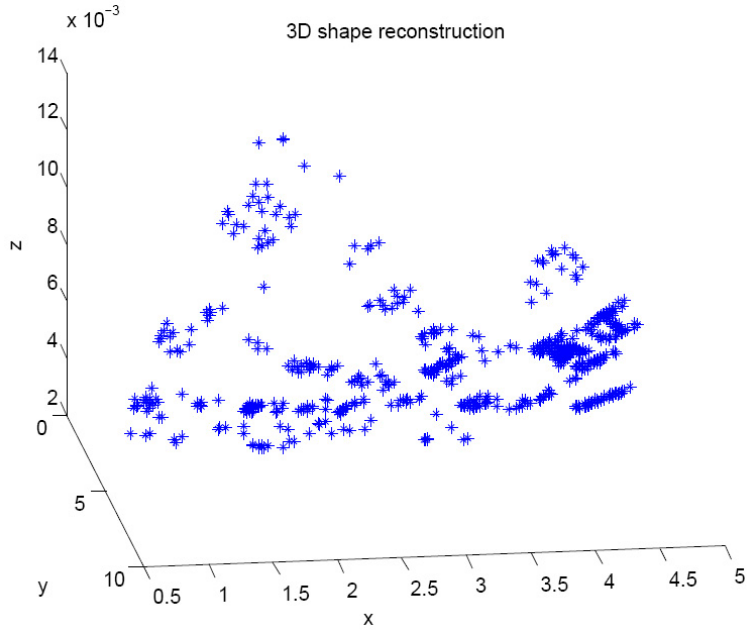


Figure 4.10: 3D shape reconstruction from the 3D points.

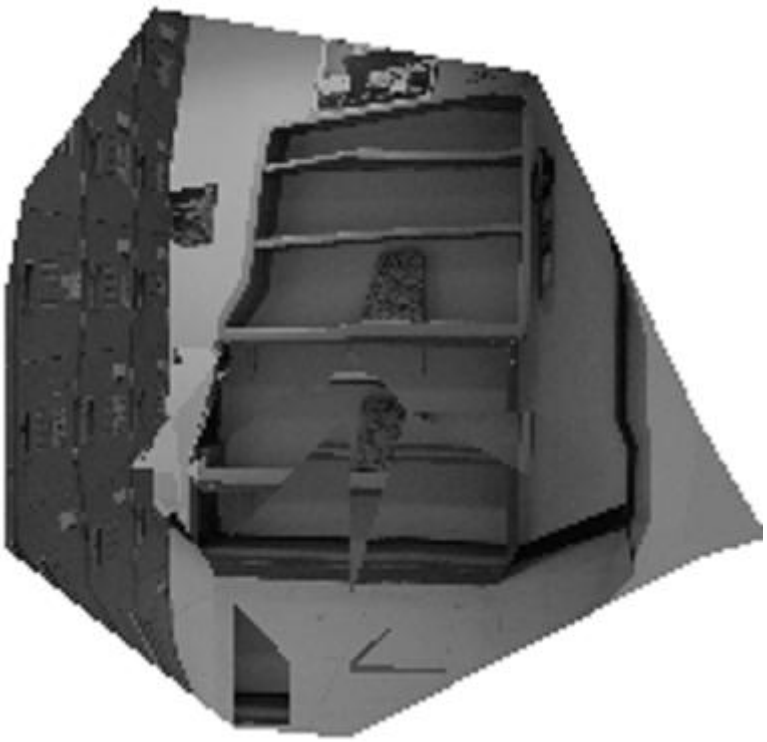


Figure 4.11: 3D texture map using Delaunay triangulation based on OpenGL.

Chapter 5

Conclusion and Recommendations

- The texture model is sufficient for human operators, guiding the robots in an indoor scenes.
- I used SIFT for features detection, for future research I recommend to use SURF.
- Due to the shortage of time, I used Delaunay triangulations for pair of images, for the future research I recommend to use it for sequence of images.
- For real time application, most part of the code should be implemented in C++ (OpenCV).
- There is a lot of research going on monocular vision, I recommend to explore it further.

References

(n.d.-a).

(n.d.-b).

Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features SURF. In (Vol. 110, pp. 346–359).

Berg, M. de, Cheong, O., Kreveld, M. van, & Overmars, M. (2008). *Computational geometry algorithms and applications* (Third Edition ed.). Springer.

Bouguet, J.-Y. (2004). *Camera calibration toolbox for matlab.* Available from: `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

Cuadros-Vargas, A., Lizier, M., Minghim, R., & Nonato, L. (2008, June). Generating segmented quality meshes from images. *Journal of Mathematical Imaging and Vision*, *33*, 11–23.

Cyganek, B., & Siebert, J. (2009). *An introduction to 3D computer vision techniques and algorithms* (First ed.). Wiley and Sons.

Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). Monoslam: Real-time single camera slam. In (Vol. 29, pp. 1052–1067).

Elfes, A. (1987, June). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, *RA-3*(3), 249–265.

Forsyth, & Ponce. (2003). *Computer vision A modern approach* (First ed.). Alan Apt.

Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (Second ed.). Cambridge University Press, ISBN: 0521540518.

Heikkila, J., & Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *Cvpr '97: Proceedings of the 1997 conference on computer vision and pattern recognition (cvpr '97)* (p. 1106). Washington, DC, USA: IEEE Computer Society.

Johnson, A. E., & Kang, S. B. (1999). Registration and integration of textured 3D data. In (pp. 234–241).

Konolige, K. (2004). Large-scale map-making.

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. In (Vol. 21, pp. 163–169). ACM, New York, NY, USA.

Lourakis, M., & Argyros, A. (2004). *The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm* (Tech. Rep.). Institute of Computer Science - FORTH.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, *60*(2), 91–110.

Ma, Y., Soatto, S., Kosecka, J., & Sastry, S. S. (2004). *An invitation to 3-D vision* (First ed.). Springer.

Moravec, J. P. (1996, september). *Robot spatial perception by stereoscopic vision and 3D evidence grids.* (Retrieved from `http://www.frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1996/9609.stereo.paper/SGabstract.html`)

Nonato, Minghim, R., Oliveira, M. C. F., & Tavares, G. (2002). A novel approach for

delaunay 3D reconstruction with a comparative analysis in the light of applications. *Computer Graphics Forum*, *20*, 161–174.

Pollefeys, M., Gool, L. V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., et al. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*, *59*(3), 207–232.

Saxena, A., Chung, S. H., & Ng, A. Y. (2008). 3-D depth reconstruction from a single still image. *International Journal of Computer Vision*, *76*.

Saxena, A., Sun, M., & Ng, A. Y. (2007). Robust software for computing camera motion parameters. *ICCV: International Conference on Computer Vision*.

Schreiber, R., Li, Z., & Baker, H. (2008). Robust software for computing camera motion parameters. *Journal of Mathematical Imaging and Vision*.

Thrun, S., Burgard, W., & Fox, D. (2000). A real-time algorithm formobile robot mapping-with applications to multi-robot and 3D mapping. In *Proceedings of the ieee international conference on robotics and automation ICRA*. San Francisco, CA: IEEE.

Trucco, E., & Verri, A. (1998). *Introductory techniques for 3-D computer vision* (First ed.). Simon and Schuster.

Tsai, R. Y., & Huang, T. S. (1984). Estimating three-dimensional motion parameters of a rigid planar patch, 111: Finite point correspondences and the three-view problem.

Uyttendaele, M., Criminisi, A., Kang, S. B., Winder, S., Szeliski, R., & Hartley, R. (2004). Image-based interactive exploration of real-world environments. In (Vol. 24, pp. 52–63).

What is structured light? (n.d.). (Retrieved from `http://www.stockeryale.com/i/lasers/structured_light.htm`)

Yanco, H., & Drury, J. (2004). "Where am I?" Acquiring situation awareness using a remote robot platform.

Zask, R. (2009). *Indoor surface model estimation and visualization using monocular camera.*