3D SCENE MODELING AND UNDERSTANDING FROM IMAGE SEQUENCES

by

Hao Tang

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2013

This manuscript has been read and accepted for the
Graduate Faculty in Computer Science in satisfaction of the
dissertation requirement for the degree of Doctor of Philosophy.

Professor Zhigang Zhu, Ph.D.

Date — Chair of Examining Committee

Professor Theodore Brown, Ph.D.

Date — Executive Officer

Professor Jizhong Xiao, Ph.D.

Professor Ioannis Stamos, Ph.D.

Dr. Rakesh Kumar, Ph.D.
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

ABSTRACT

3D SCENE MODELING AND UNDERSTANDING FROM IMAGE SEQUENCES

by

Hao Tang

Adviser: Professor Zhigang Zhu

A new method for 3D modeling is proposed, which generates a content-based 3D mosaic (CB3M) representation for long video sequences of 3D, dynamic urban scenes captured by a camera on a mobile platform. In the first phase, a set of parallel-perspective (pushbroom) mosaics with varying viewing directions is generated to capture both the 3D and dynamic aspects of the scene under the camera coverage. In the second phase, a unified patch-based stereo matching algorithm is applied to extract parametric representations of the color, structure and motion of the dynamic and/or 3D objects in urban scenes, where a lot of planar surfaces exist. Multiple pairs of stereo mosaics are used for facilitating reliable stereo matching, occlusion handling, accurate 3D reconstruction and robust moving target detection. The outcome of this phase is a CB3M representation, which is a highly compressed visual representation for a dynamic 3D scene, and has object contents of both 3D and motion information. In the third phase, a multi-layer based scene understanding algorithm is proposed, resulting in a planar surface model for higher-level object representations. Experimental results are given for both simulated and several different real video sequences of large-scale 3D scenes to show the accuracy and effectiveness of the representation. We also show the patch-based stereo matching algorithm and the CB3M representation can be generalized to 3D modeling with perspective views using either a single camera or a stereovision head on a ground mobile platform or a pedestrian. Applications of the proposed method include airborne or ground video surveillance, 3D urban scene modeling, traffic survey, transportation planning and the visual aid for perception and navigation of blind people.

ACKNOWLEDGMENTS

I would like to thank Professor Zhigang Zhu for his guidance and support throughout my graduate studies. He gave me the opportunity to work in CCVCL as soon as I started my research in computer vision. He provided an exciting working environment with many opportunities to develop new ideas, work on promising applications and meet many interesting people. I would also like to thank the other members of my thesis committee: Professor Jizhong Xiao, Professor Ioannis Stamos, and Dr. Rakesh (Teddy) Kumar, for their comments and feedbacks.

I am grateful to Professor George Wolberg for introducing me to research area of image processing and for encouraging me to pursue graduate studies. I also want to thank Professor Ted Brown, Executive Officer of the CUNY CS PhD Program, for the both financial and academic supports throughout my PhD studies.

I want to thank my lab mates and colleagues at CCVCL. I am very happy to have become good friends with Weihong Li, Edgardo Molina, Tao Wang, Wai Khoo, and Yufu Qu. I have enjoyed every moment with them.

I would like to give special thanks to Teddy Kumar and Supun Samarasekera for providing research opportunities and mentorship during my three year internships at SRI Sarnoff. During the three-year research internship at Sarnoff, I have benefited from interactions with many people. I would like to thank Charles Karney, Garbis Salgian, Targey Oskiper, Yi Tan, Han-pang Chiu, Chao Zhang and Bing-bing Chai for many illuminating discussions.

The work has been supported by National Science Foundation (NSF) under Award #EFRI-1137172 and Award #CNS-0551598, Air Force Research Laboratory (AFRL) under Award No. FA8650-05-1-1853, and Army Research Office (ARO) under Award #W911NF-08-1-0531. These supports have greatly inspired and facilitated the research reported in my PhD thesis.

Finally, I am deeply grateful to my parents and brother, for their love and support. My heartfelt appreciation goes to my wife Ling, and my two lovely children, Vicky and Victor. This thesis is dedicated to them.

TABLE OF CONTENTS

Abstract iv

Acknowledgments v

Table of Contents vi

Lists of Tables viii

Lists of Figures ix

Chapter 1 Introduction 1

1.1 Problem Statement 1

1.2 Thesis Statement and Contributions 2

Chapter 2 Related Work 6

2.1 Mosaics in Parallel-perspective Geometry 6

2.2 3D Modeling from Stereo Matching of Video 7

2.3 3D Modeling from Video Mosaics 8

2.4 Simultaneous Localization And Mapping (SLAM) 9

Chapter 3 3D Modeling from Image Sequence - a Mosaic based Method Using Pushbroom Geometry 11

3.1 Dynamic Pushbroom Stereo Mosaic Geometry 12

3.2 Real-World Issues and Multi-View Mosaics 16

3.3 3D and Motion Content Extraction 21

3.4 CB3M: Content-Based 3D Mosaics 32

3.5 Experimental Results and Analysis 36

Chapter 4 Scene Understanding from Content Based 3D Mosaics 46

4.1 Surface Layer Generation 46

4.2 Structural Layer Generation 51

4.3 Cluster Layer Generation 52

4.4 Summary and Discussions 53

Chapter 5 3D Modeling from Image Sequence – Using Perspective Geometry 55

5.1 3D Modeling from Stereo Image 55

5.2 Smart Sampling 60
5.3 Experimental and Results 64
5.4 Summary and Discussions 66
Chapter 6 Conclusions 68
6.1 Summary 68
6.2 Future Work 69
Appendix A: Performance Evaluation of CB3M in Simulation Scene 72
Appendix B: Computation Time Analysis in Both stereo Mosaicing and Content Extraction 76
BIBLIOGRAPHY 78

LISTS OF TABLES

Table 1. Work flow of the generation of the surface layer 49

Table 2. Comparison of average depth estimation errors: two views and multiple views 73

Table 3. Motion estimation errors 74

Table 4. Computation time analysis 77

LISTS OF FIGURES

Figure 1-1 System diagram 3

Figure 3-1 Dynamic pushbroom stereo mosaics 13

Figure 3-2 Ray interpolation for a dynamic scene 18

Figure 3-3 Multi-view pushbroom mosaics 20

Figure 3-4 Height from dynamic pushbroom stereo: (a) an infeasible pair; (b) a feasible pair 21

Figure 3-5 Natural matching primitives 24

Figure 3-6 An example of region matching results. The matches are marked as "X", with corresponding colors. 26

Figure 3-7 An example of surface fitting results. Both the mismatch and the small error in the initial match are fixed 26

Figure 3-8 Content-based 3D mosaic representation 33

Figure 3-9 (a) The leftmost, (b) center and (c) rightmost views of the nine mosaics of a simulated scene. The final CB3M representation is shown in (d). Each region is rendered by its average color. Plane parameters *(a,b,c,d)* (in blue) and boundaries for several representative surfaces, and motion displacements (s_x, s_y) (in red) of the detected moving targets are labeled in (d). For comparison, (e) and (f) show the rendered "height" maps of the scene from the stereo matching results from the 1st stereo pair only, and from all the mosaics, respectively. Finer and more accurate results are obtained in (f). Regions marked in red are the "outliers" that will be passed to the moving target test; some of them are due to occlusions at depth boundaries rather than independent motion, but they are too thin or too small to be a real moving targets. The detected moving targets are shown in (d). 37

Figure 3-10. 3D and motion from multi-view stereo mosaics of an aerial video sequence. (a) A pair of stereo mosaics from the total nine mosaics;'' (b) height map of entire mosaic; (c) close-up of the 1st window marked in (a); and (d) the height map of the objects inside that window, with the detected moving targets marked by their boundaries and those not detected by rectangular boxes; (d) close-up of the 2nd window marked in (a); and (f) the height map of that window. 39

Figure 3-11. Content-based 3D mosaic representation of an aerial video sequence. Only a window is shown, with some of the regions labeled by their boundaries and plane parameters (in blue), and the detected moving targets marked by their boundaries and motion vectors (in red). 40

Figure 3-12. A 4816 (W) x 2016 (H) mosaic from a 758-frame high-resolution NYC video sequence. The Manhattan world geometric constraint is illustrated on the mosaic 42

Figure 3-13. (a) Depth from a pair of mosaics, (b) from four mosaics, and (c) color-coded depth map of (b) 43

Figure 3-14. Moving target detection using the road direction constraint. In the figure (a) and (b) are the corresponding color images and height maps of the 1st (bottom-left) and 2nd (top-right) windows in Fig. 3-12, with the detected moving targets painted in red. The two circles show the three moving targets that are not detected. The arrows indicate the directions of the roads along which the moving targets are searched..... 44

Figure 4-1. The original mosaic overlaid by patches without enough confidences (marked in light blue). Most of them are on the boundaries of buildings 50

Figure 4-2. Surface layer generation. Patches shared with similar geometry are labeled into the same color on the boundaries. Most of patches on the ground plane are marked in same label (pink) after surface layer though a few of them are labeled in green, yellow and brown. All colors are randomly selected. 50

Figure 4-3. Star-shaped graph representation of building, the center node is building top and rest are façade, edges in the graph means two connected nodes are neighbor and perpendicular. The dashed lines represent the connections between two neighbor surfaces a not required. 51

Figure 4-4, Structural layer generation. Patches on building are labeled in red. Note that how the roof and facades of e ach building is mostly labeled into one structure..... 52

Figure 4-5. Cluster layer generation. Patches attached to buildings are labeled in yellow. 53

Figure 5-1. (a) the segmentation result of a reference image; (b) a close-up window shows more interest points are extracted in vertical boundary. 57

Figure 5-2. (a) The simulated scene with a number of objects (a pole is in a close range); (b) 3D depth map of the simulated scene; (c) sampling results of the 2D image and 3D depth map using a regular

sampling method; Note the pole is missing after the regular sampling; (d) sampling results of the 2D image and 3D depth map using our smart sub-sampling method; the pole is still preserved after sampling. 62

Figure 5-3. (a) The sampled depth image using background removal method; (b) The sampled original image using background removal method (note: the pole and the object are kept in (a) and (b)); (c) The sampled original image using the image re-illumination; (d-f) the object of interest (the pole) in closed range is shifted left and right to simulate its motion parallax. 63

Figure 5-4. (a) Reference image (color image in left view); and (b) 3D depth map generated by patch-based method (the brighter, the closer). For several large regions indexed, the boundaries of regions are marked by closed curves (blue) and planar parameters are drawn on the regions: each arrow and the numbers in a pair of parentheses represent the surface norm, and last number (meters) represents the distance from the surface center to the camera. 65

Figure 5-5. (a) A pair of stereo images of an indoor scene captured in an office with a number of objects (note: a tripod is in a close range); (b) 3D depth map of the indoor scene; pixels with large uncertainty are marked in green; (c) sampling results of 2D image and 3D depth map using uniform sampling method: the tripod is missing after regular sampling; (d) sampling results of 2D image and 3D depth map using smart sampling method: the tripod is kept after sampling; (e) sampling results of highlighting objects close to the user, after removing background and applying image dilation; (f) sampling results using motion parallax simulation and only an OI close to the user is shifted towards left, kept original position and shifted towards right, respectively. 65

Figure A-1. Depth error analysis. (a) Error histogram. (b) Comparison and selection among the results from the 8 pairs of stereo mosaics for the largest 17 regions. The last column (9th) shows the final selection. 73

Chapter 1 INTRODUCTION

Reconstructing and representing large-scale 3D scenes from image sequence have many important applications, including airborne or ground video surveillance for moving target extraction, automated 3D urban scene construction, airborne/ground traffic survey, and image-based modeling and rendering.

For these applications, there are two major challenges. First, hours of video streams may be generated every time the mobile platform performs a data collection task. The data amount is in the order of 100 GB per hour for standard 640*480 raw color images. The huge amount of video data not only poses difficulties in data recording and archiving but also is prohibitive for users to retrieve, review or to process. Second, efficient and accurate reconstruction 3D model for image sequences is a difficult problem too. It also requires robust and accurate camera orientation estimation for many video frames.

In order to cope with the above difficult problems, Zhu, et al, (2004) proposed a mosaic based method. Given a video sequence captured by a moving platform, a set of parallel-perspective (pushbroom) mosaics with varying viewing directions is generated, so huge amount of data turns to several panoramas covering same large field of view as the original video. In that method, parallel ray interpolation for stereo mosaicing (PRISM) that can generates seamless mosaic under motion parallax and 3D models can be reconstructed based on the stereo mosaics. There are a number of advantages of the above method. First, a set of the multi-view pushbroom mosaics, with a pair of stereo mosaics as the minimum sub-set, is a compact visual representation for a long video sequence of a 3D scene and it allows efficient transmission between moving platform and backend processing server. Second, in a theoretical analysis (Zhu, et al, 2003), with parallel-perspective stereo mosaic, that depth error is constant in theory and is linearly proportional to depth in practice. Zhu, et al, (2005) implemented a practical method in camera orientation estimation with external orientation measurements; however 3D reconstruction and dynamic moving object extraction were not dealt with, which open up many research issues.

1.1 PROBLEM STATEMENT

We are interested in 3D dynamic urban scene modeling and understanding, which finds many applications in recent years in urban surveillance, traffic management, urban planning and entertainment.

Although the above method gives a feasible solution for the reconstruction and representation of a large-scale urban scene, there are a number of important problems we would like to investigate.

First, due to the 3D nature of urban scene observed by a moving platform, could obvious motion parallax and object occlusions be effectively handled in order to detect moving objects of interest? Most of the existing algorithms using change detection assuming planar scene or stationary camera will fail in this situation.

Second, can we design a unified approach for both accurate 3D modeling and effective moving target detection, for large-scale 3D man-made urban scenes with fine structures, textureless regions, sharp depth changes, and occlusions?

Third, can the context information about both the static objects (buildings, roads and facilities) and moving targets be encoded in a highly compressed form?

Fourth, assume an accurate and compact reconstruction and representation are able to obtained, can these be used for a higher-level scene understanding, as well as for the refinement of the generated 3D modeling?

1.2 THESIS STATEMENT AND CONTRIBUTIONS

With the above questions, based on the previous effort made by Zhu, et al, (2004, 2003, 2005), we have made the following five major new contributions (Fig. 1-1).

- First, we extend the previous work on stereo mosaics from static scenes to dynamic scenes, thus allowing the handling of independent moving objects. This is significant in low-altitude aerial video surveillance of urban scenes since traditional methods using change detection fail to work due to motion parallax. We also show that the PRISM algorithm also works for dynamic scenes, which means we can re-use the code we have developed for stereo mosaics of static scenes. These results are presented in Section 3.1 and Section 3.2.

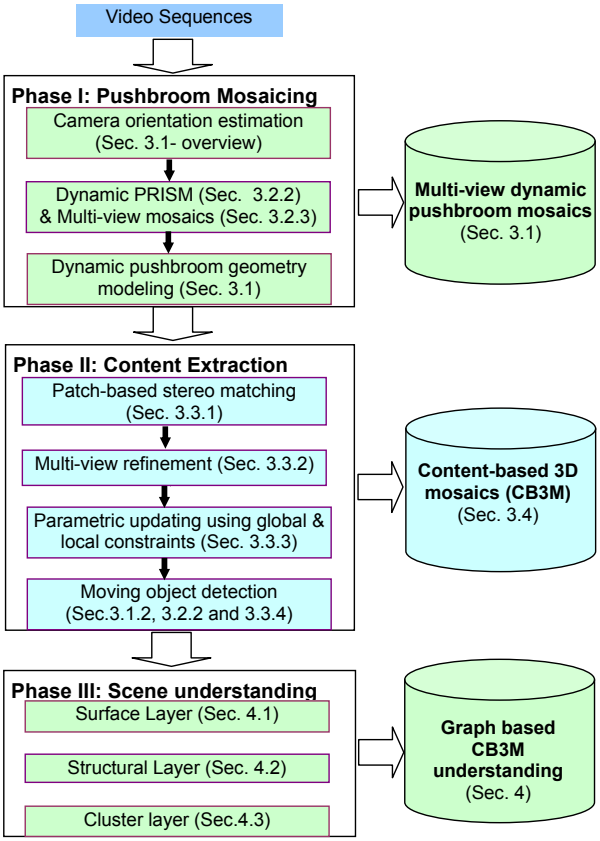


Figure 1-1. System diagram

- Second, an effective and efficient patch-based stereo matching algorithm has been proposed to extract both 3D and motion information from stereo mosaics of urban scenes, which feature sharp depth boundaries and many textureless regions. This is a unified approach for both 3D reconstruction and moving target extraction. Furthermore, this method can produce higher-level scene representations rather than just depth maps, which leads to our highly compressed content-based video representation. The algorithm will be presented in Section 3.3.

- Third, we perform thorough experimental analysis of the robustness and accuracy of 3D reconstruction using parallel-perspective stereo mosaics. We show the high accuracy of 3D reconstruction and moving target detection by using a simulated video sequence while both ground truth data of 3D urban model and accurate camera orientation information are available, which motivates us and other researchers for developing robust and efficient algorithms to

estimate camera orientation with many image frames. On the other hand, using a simplified camera orientation estimation method for several real-world video sequences, we have found that we can generate very compelling stereo perception and reliable 3D depth information. This indicates that for some applications where accurate 3D measurements are not critical, such as image-based rendering, and even automatic target detection and transportation analysis, we can ease the challenging problem of many-frame camera orientation estimation. The experimental analysis is discussed in Section 3.5.

- Fourth, a content-based 3D mosaic (CB3M) representation is proposed (Section 3.4), and a graph-based higher-level scene understanding algorithm is proposed (Chapter 4). The method shows the advantage of CB3M data representation in grouping planar patches into higher-level object entities, such as roads, buildings, etc., and in inferring context information. Based on the graph-based 3D planar surface model, other geometric and texture information, further scene understanding can be achieved. Some preliminary experiments on simulated data are presented in Chapter 4; the algorithm will be further developed and tested for real-world video sequences.
- Finally, the new patch-based stereo matching algorithm can also be used with other stereo geometry, such as perspective stereo (Chapter 5). With the patch based stereovision approach under perspective geometry, we propose the smart sub-sampling to transduce the important/highlighted information, and/or remove background information, before presenting to visual impaired people. The patch-based method plays an important role in the smart sub-sampling step: it generates surface based 3D depth map instead of 3D point clouds, therefore, it carries more meaningful information and it's easy to convey these information to visual impaired people.

The rest of the dissertation is organized as follows. Chapter 2 discusses some related work, including stereo mosaicing approaches, 3D from traditional stereo using video, 3D from stereo mosaics, and simultaneous localization and mapping (SLAM) with video cameras.

In Chapter 3, the mathematic model and research topics and methodology are discussed. The mathematical framework of the dynamic pushbroom stereo and its properties for moving target extraction

in Section 3.1. In Section 3.2, technical issues of dynamic stereo mosaics in real-world applications are discussed, and multi-view pushbroom mosaics are proposed for image-based rendering and for extracting 3D structure and moving targets. In Section 3.3, our multi-view pushbroom stereo matching algorithm for 3D reconstruction and moving target extraction is provided. Then in Section 3.4, a content-based 3D mosaic representation (CB3M) is described. Experimental results of CB3M representation construction will be given in Section 3.5 with both simulated and several very different video sequences.

In Chapter 4, based on the CB3M data, a graph based multi-layer scene understanding method is proposed and experiment on the simulation data is given.

In Chapter 5 the patch based 3D modeling algorithm applying to the perspective projection geometry is discussed and it is applied in the application of visual prosthesis.

Finally, in Chapter 6, a conclusion is drawn and some proposed directions of future research will be discussed.

Chapter 2 RELATED WORK

Reconstructing and representing large-scale 3D scenes from multiple images has attracted a lot of attention. For example, the work at CMU (Herman & Kanade, 1984; Herman & Kanade, 1986) represents one of the first efforts in incrementally constructing 3D scenes from multiple complex images. Interestingly, they used "3D MOSAIC" as the name of their system. However, it is the advancement of both hardware and software in the last two decades that makes it possible to efficiently process huge amounts of video data and to generate panoramic mosaics using a general purpose PC. We will briefly discuss four groups of work that are related to this thesis: mosaics in parallel-perspective geometry, 3D modeling from stereo matching with video sequences, 3D modeling from video mosaics, and simultaneous localization and mapping (SLAM) with video cameras.

2.1 MOSAICS IN PARALLEL-PERSPECTIVE GEOMETRY

Mosaics have become common for combining and representing a set of images gathered by one moving camera or multiple cameras. In the past, video mosaic approaches (Irani, et al, 1996; Hsu & Anandan, 1996; Odone, et al, 2000; Leung & Chen, 2000, Cai et al, 2010, Vivet et al. 2011) have been proposed for video representation and compression, but most of the work is for generating 2D mosaics instead of 3D panoramas, and using panning (rotating) cameras for arbitrary scenes or moving cameras for planar scenes, instead of traveling (translating) cameras typically used in airborne or ground mobile urban surveillance and 3D scene modeling. In the latter applications, obvious motion parallax is the main characterization of the video sequences due to the self-motion of the sensors and obvious depth changes of the scenes.

To generate truly "3D mosaics" from video sequences of a traveling camera, we are particularly interested in the parallel-perspective pushbroom stereo geometry (Chai & Shum, 2000; Zhu, et al, 2004). The term "pushbroom" is borrowed from satellite pushbroom imaging (Gupta & Hartley, 1997) where a linear pushbroom camera is used. The basic idea of the pushbroom stereo mosaics is as follows. If we assume the motion of a camera is a 1D translation and the optical axis is perpendicular to the motion, then we can generate two spatio-temporal images (mosaics) by extracting two scanlines of pixels of each frame

(perpendicular to the motion of the camera), one in the leading edge and the other in the trailing edge. Each mosaic image thus generated is similar to a parallel-perspective image captured by a linear pushbroom camera, which has parallel projection in the direction of the camera's motion and perspective projection in the direction perpendicular to that motion. Pushbroom stereo mosaics have uniform depth resolution, which is better than with perspective stereo, and the multi-perspective stereo with circular projection (Peleg, et al 2001; Shum & Szeliski, 1999). Pushbroom stereo mosaics can be used in applications where the motion of the camera has a dominant translational direction. Examples include satellite pushbroom imaging (Gupta & Hartley, 1997), airborne video surveillance (Zhu, et al, 2004), image-based rendering with 3D reconstruction or 3D estimation (Chai & Shum, 2000, Rav-Acha, et al, 2008), 3D representations of ground route scenes (Zheng & Tsuji, 1992; Zhu & Hanson, 2004, Zheng & Shi, 2008), under-vehicle inspection (Dickson, et al, 2002; Koschan, et al, 2004), 3D measurements of industrial parts by an X-ray scanning system (Noble, et al, 1994), and 3D gamma-ray cargo inspection (Zhu & Hu, 2007). Some work has been done in 3D reconstruction of panoramic mosaics (Li, et al, 2004; Sun & Peleg, 2004) with an off-center rotation camera, but the methods are limited to a fixed view-point camera instead of a moving camera, and usually the results are still low-level 3D depth maps of static scenes, instead of high-level 3D structural representations for both static and dynamic target extraction and indexing. On the other hand, layered representations (e.g., Xiao & Shah, 2004; Zhou & Tao, 2003; Ke & Kanade, 2001) have been studied for motion sequence representations; however, the methods are usually computationally expensive, and the outputs are typically motion segmentation represented by affine planes instead of true 3D information. Efficient, high-level, content-based, and very low bit-rate representations of videos of 3D scenes and moving targets are still in great demand.

2.2 3D MODELING FROM STEREO MATCHING OF VIDEO

Another class of related work is 3D reconstruction from stereo pairs. Stereo vision is one of the most important topics in computer vision, and a thorough comparison study (Scharstein & Szeliski, 2002) has been performed. Simple window-based correlation approaches do not work well for man-made scenes. In the past, an adaptive window approach (Kanade & Okutomi, 1991) and a nine-window approach (Fusiello, et al, 1997) have been used to deal with some of these issues. Later on, color segmentation has been used for refining an initial depth map to get sharp depth boundaries and to obtain depth values for

textureless areas (e.g., Tao, et al, 2001), and for accurate layer extraction (e.g., Ke & Kanade, 2001). Global optimization based stereo matching methods, such as belief propagation (Sun, et al, 2003) and graph cuts (Boykov, et al, 2001; Kolmogorov & Zabih, 2001), can obtain accurate depth information, but these methods are computationally expensive. Cornelis, et al (2008) present a complete system for turning forward-looking stereo video from a moving car into a model from which a virtual drive-through of a city street can be rendered. The paper by Pollefeys, et al (2008) describes a system for automatic, geo-registered, real-time 3D reconstruction from video of urban scenes using a multi-view stereo approach. Most stereo reconstruction papers are based on perspective stereo geometry, except a few papers (Li, et al, 2004; Sun & Peleg, 2004; Zhu & Hanson, 2004) dealing with multi-perspective stereo images.

2.3 3D MODELING FROM VIDEO MOSAICS

The two works which are the most similar to ours are (Rav-Acha et al. 2008) and (Zheng and Shi 2008). Both generate large panoramas and 3D models from video. Rav-Acha, et al (2008) proposed a Minimal Aspect Distortion (MAD) mosaicing method that uses depth to minimize the geometrical distortions of long panoramic images, because pure 2D image based stitching cannot avoid distortions in mosaic generation. One should realize that only depth changes cause distortion, while understanding where depth changes and then trying to keep perspective projection in the areas of changing depths is a solution to minimize distortions. Therefore, the method first reconstructs the 3D of video sequence (by computing the camera motion and local 3D maps), and then generates 2D mosaics.

Zheng and Shi (2008) first generate parallel-perspective mosaics and then reconstruct a 3D model. A camera is mounted on a vehicle running along with a straight road and camera is facing to the side of street. A parallel-perspective panorama is first generated by scanning 1D image over the scene. The camera motion is simple and can be estimated by GPS. In order to avoid error-prone feature matching to reconstruct the corresponding 3D map of the mosaic, the paper proposes a method to measure depth information using image velocity that is a ratio of the spatial and temporal difference in the spatiotemporal space (x-y-t space). Since the object closed to camera is ‘moving’ fast in the panorama and should have strong contrast difference (measuring by check image gradient along both spatial and temporal space), the depth information is computed only at points with high spatial-temporal gradients, then a full depth

map can be filled, assuming depth changes only occur on location with strong gradients (edges). One of the motivations is that 3D recovery is challenging for street scenes, since they contain many windows and other reflective surfaces causing error-prone matches. Although the recovered depth information may not be precise, the system can avoid the heavy computation of 3D reconstruction and may offer real time 3D modeling.

Both work particular target to generate seamless panoramas for the static scene in a ground based application, while the method described in this dissertation can deal with both static and dynamic scenes and in both ground based and aerial based applications.

2.4 SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

Pose estimation is a key step to 3D modeling. Solutions can be classified into two groups: Simultaneous Localization and Mapping (SLAM) (Davison and Murray 2001; Davison et al. 2007, Oskiper et al. 2007, Doucet and Johansen 2008), and Structure form Motion (SFM) (Nistér et al. 2004, Mouragnon et al. 2006, Klein and Murray 2007, Newcombe and Davision 2010). SLAM is first proposed in robot field around mid-1980s (Smith, and Cheeseman 1986). The topic addresses an important problem in robotics: to build up a map in an unknown environment meanwhile keeping track of a robot's current position. It is typically treated as the problem of estimating spatial uncertainties of both scene structure and the robot's location, and is usually modeled in a probabilistic framework. On the other hand, classical structure from motion (SFM) methods solve the "mapping and localization" problem by using iterative global optimization methods (bundle adjustments) that minimizes the overall re-projection errors of sparse features among an image sequence (Hartley and Zisserman 2000). Recently these two classes of methods (SLAM and SFM) are coming together when the main sensors are cameras; many solutions have been provided by combining the above two groups of methods (e.g., Mouragnon 2006, Klein and Murray 2007).

Recent advances of vision algorithms and hardware enable design and implementation of real time visual navigation system. However, algorithms using pure vision methods still face the well-known problems (i.e. drift and break), though systems using stereo camera can produce relative reliable results. One general solution relies on a global optimization method, either locally or globally. However, solving the problem in real-time for a large environment is not allowed due to the expensive computation of global optimization.

Parallel tracking and mapping (Klein and Murray 2007) partially solves the problem, but it's still limited in a small working environment. Therefore, incorporating probabilistic framework gives an alternative solution to build a more reliable system. In other words, the Kalman filter is beneficial when processing time is limited, otherwise using bundle adjustment can gives an optimized solution. This is shown by the work of Strasdat, et al. (2010). Hardware improvement or hardware speedup may enable real time visual odometry systems using global optimization to produce reliable and accurate results in the near future.

Chapter 3 3D MODELING FROM IMAGE SEQUENCE - A MOSAIC BASED METHOD USING PUSHBROOM GEOMETRY

In this dissertation we address the problems of visual representations for large amounts of video stream data, of dynamic three-dimensional (3D) urban scenes, captured by a camera mounted on a low-altitude airborne or a ground mobile platform.

In Chapter 3, we describe a content-based 3D mosaic representation (CB3M) for video sequences of 3D and dynamic scenes captured by such a camera mounted on a mobile platform. This chapter includes work in the first two phases in Fig. 1-1. The motion of the camera has a dominant direction of motion (as on an airplane or ground vehicle), but 6 DOF motion is allowed. We have developed a three-phase procedure for this goal, as shown in Fig. 1-1. In the first phase, a set of parallel-perspective (pushbroom) mosaics with varying viewing directions is generated to capture both the 3D and dynamic aspects of the scene under the camera coverage. Bundle adjustment techniques can be used for camera pose estimation, sometimes integrated with the geo-referenced data from GPS and INS when available. A ray interpolation approach called PRISM (parallel ray interpolation for stereo mosaicing) is used to generate multiple seamless parallel-perspective mosaics under the obvious motion parallax of a translating camera. The set of the multi-view dynamic pushbroom mosaics, with a pair of stereo mosaics as the minimum sub-set, is a compact visual representation for a long video sequence of a 3D scene with independent moving targets. In this phase, the epipolar geometry of the multi-perspective pushbroom stereo mosaics is also established to facilitate stereo matching and moving target detection in the next phase.

However, the 2D mosaic representation is still an image-based one without object content representations. Therefore, in the second phase, a segmentation-based (“patch-based”) stereo matching approach is proposed to extract parametric representation of the color, structure and motion of the dynamic and/or 3D objects (i.e., the contents) in urban scenes, where a lot of planar surfaces exist. In our approach, we use the fact that all the static objects obey the epipolar geometry, i.e. along the epipolar lines of pushbroom stereo. An independent moving object (moving on a road surface), on the other hand,

either violates the epipolar geometry if the motion is not in the direction of sensor motion, or exhibits unusual 3D structure otherwise, e.g., obviously hanging above the road or hiding below the road. Furthermore, multiple pairs of stereo mosaics and local/global spatial constraints are used for facilitating reliable stereo matching, occlusion handling, accurate 3D reconstruction and robust moving target detection.

Based on the above two phases, a content-based 3D mosaic (CB3M) representation is created for a long video sequence. This is a highly compressed visual representation for the video sequence of a dynamic 3D scene. For example, a real image sequence of a campus scene has 1000 frames of 640*480 color images. With its CB3M representation, a compression ratio of more than 10,000 is achieved. More importantly, the CB3M representation has high-level object contents. A scene is represented in parametric forms of planar regions with their 3D, their boundaries, their motion, and their relations. The CB3M representation can be utilized for object recognition and indexing.

In this chapter, the basic mathematic model of dynamic pushbroom stereo geometry is discussed, then the unified patch based 3D modeling algorithm is described. After the introduction of the Content Based 3D Mosaic (CB3M) representation, detailed experimental results, in both simulation and real data, are provided.

3.1 DYNAMIC PUSHBROOM STEREO MOSAIC GEOMETRY

Stereo mosaics of static scenes have been well-studied in the past. As a preparation, we give a brief description of the concept. Assume the motion of a camera is an ideal 1D translation, the optical axis is perpendicular to the motion, and the frames are dense enough. Then, we can generate two spatio-temporal images by extracting two columns of pixels (perpendicular to the motion) at the leading and trailing edges of each frame in motion. The geometry in this ideal case (i.e. 1D translation with constant speed) is the same as the linear pushbroom camera model (Gupta & Hartley, 1997). Therefore we also call this image representation pushbroom stereo mosaic representation. A generalized model under 3D translation (Zhu, et al 2004) has extended the parallel-perspective stereo geometry to image sequences with 3D translation and further with 6 DOF motion (rotation + translation). Here, we will use the parallel-

perspective stereo geometry under 1D translation to introduce the new concept of the dynamic stereo mosaics.

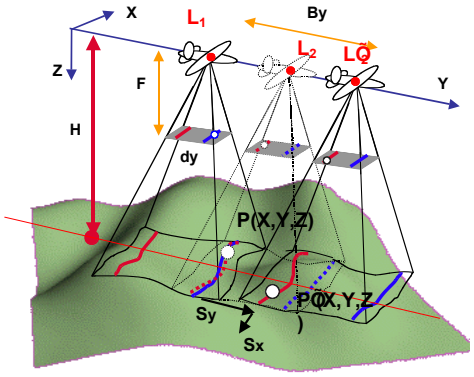


Figure 3-1 Dynamic pushbroom stereo mosaics

3.1.1 Dynamic pushbroom stereo model

For completeness, we start with the formulation of the pushbroom stereo mosaics in a static scene. Without loss of generality, we assume that two slit windows of two scanline locations have d_{yl} and d_{yr} offsets to the center of the image, respectively, and the distance between the two windows is the fixed "disparity" $d_y = d_{yl} - d_{yr} > 0$ (in Fig. 3-1, $d_{yl} = d_y/2$, $d_{yr} = -d_y/2$). The "left eye" view (x_l, y_l) is generated from the front slit window d_{yl}, while the "right eye" view (x_r, y_r) is generated from the rear slit window d_{yr}. A static point P (X,Y,Z) can be viewed twice from the two slit windows, at the camera location L_1 and L_2, respectively. Then the *parallel-perspective" pushbroom" model* of the stereo mosaics thus generated can be represented by

$$\begin{cases} x_l = x_r = F\dfrac{X}{Z} \\ y_l = F\dfrac{Y}{H} - (\dfrac{Z}{H} - 1)d_{yl} \\ y_r = F\dfrac{Y}{H} - (\dfrac{Z}{H} - 1)d_{yr} \end{cases} \qquad (3\text{-}1)$$

where F is the focal length of the camera, H is the height of a *fixation plane* on which we want to align our stereo mosaics. Eq. (3-1) gives the relation between a pair of 2D points, (x_l, y_l) and (x_r, y_r), one from each mosaic, and their corresponding 3D point P (X,Y,Z). It serves a function similar to the classical pin-hole perspective camera model. From Eq. (3-1) the depth of the point P can be computed as

$$Z = H\frac{b_y}{d_y} = H(1+\frac{\Delta y}{d_y}) \quad (3\text{-}2)$$

where $b_y = d_y + \Delta y = F\frac{B_y}{H}$ is the "scaled" version (in pixel) of the "baseline" B_y, i.e., the distance between two camera locations, and

$$\Delta y = y_r - y_l \quad (3\text{-}3)$$

is the "mosaic displacement" in the stereo mosaics. We use "displacement" instead of "disparity" since it is related to the baseline in a two view-perspective stereo system. Displacement Δy is a function of the depth variation of the scene around the fixation plane H. Since a fixed angle between the two viewing rays is selected for generating the stereo mosaics, the "disparities" (d_y) of all points are fixed; instead geometry of optimal/adaptive baselines (b_y) for all the points is created. In other words, for any point in the left mosaic, searching for the match point in the right mosaic means (virtually) finding an original image frame in which the match pair has a pre-defined disparity (by the distance of the two slit windows) and hence has an adaptive baseline depending on the depth of the point. Therefore, a stereo geometry with uniform depth resolution is achieved. More in-depth analysis on depth accuracy of stereo mosaics from real image sequences can be found in Zhu, et al, 2003. Here we focus more on the dynamic aspect of stereo mosaics, and algorithms for simultaneous 3D reconstruction and moving target detection in urban scenes.

Interestingly, dynamic pushbroom stereo mosaics are generated in the same way as with the static pushbroom stereo mosaics described above. Fig. 3-1 also illustrates the geometry. A 3D point P (X,Y,Z) on a target is first seen through the leading edge (the front slit window) of an image frame when the camera is at location L_1. As we have discussed, if the point P is static, we can expect to see it through the trailing edge (rear slit window) of an image frame when the camera is at location L_2. However, if the point P moves during that time, the camera needs to be at a different location L'_2 to see this moving point through its trailing edge. To simplify the equations, we assume that the motion of the moving point between two observations (L_1 and L'_2) is a 2D motion (S_x, S_y), which implies that the depth of the point does not change over that period of time. Therefore, the depth of the moving point can be calculated as

$$Z = F\frac{B_y - S_y}{d_y} \tag{3-4}$$

where B_y now is denoted as the distance of the two camera locations (L_1 and L'_2 in the y direction). Mapping this relation into the stereo mosaic notation above (Eq. 3-2), we have

$$Z = H(1 + \frac{\Delta y - s_y}{d_y}) \tag{3-5}$$

and

$$(S_x, S_y) = (Z\frac{s_x}{F}, H\frac{s_y}{F}) = (Z\frac{\Delta x}{F}, H\frac{s_y}{F}) \tag{3-6}$$

where (Δx, Δy) is the visual motion of the moving 3D point P, which can be measured in the stereo mosaics. The vector (s_x, s_y) is the target motion represented in stereo mosaics. Obviously, we have $s_x = \Delta x$. The above analysis only shows the geometry of a moving camera with 1D translational motion. A pair of generalized stereo mosaics can be generated when the camera undertakes constrained 6 DOF motion, similar to the case of static scenes (Zhu, et al, 2004).

3.1.2 Moving object extraction against parallax

We have the following interesting observations about the *dynamic* pushbroom stereo geometry for 3D and moving target extraction when obvious motion parallax exists in videos of 3D urban scenes.

(1) *Stereo fixation.* For a static point (i.e. $S_x = S_y = 0$), the visual displacement of the point with a depth H is (0, 0), indicating that the stereo mosaics thus generated fixate on the plane of depth H. If the fixation plane is the ground plane, this fixation facilitates stereo matching and moving target detection since the major background (i.e., the ground plane) has been aligned.

(2) *Motion accumulation*. For a moving point ($S_x \neq 0$ and/or $S_y \neq 0$), the motion between two observations accumulates over a period of time due to the large distance between the leading and trailing edges in creating the stereo mosaics. This will increase the discrimination capability for slowly moving objects viewed from a relatively fast moving aerial camera. Typically, a moving object as recorded in a pair of stereo mosaics is originally viewed from two views that are many frames apart (Fig. 3-1).

(3) *Epipolar constraints.* In the ideal case of 1D translation of the camera (with which we present our dynamic pushbroom stereo geometry in this paper), the correspondences of static points are along horizontal epipolar lines in a pair of pushbroom mosaics, i.e., $\Delta x = 0$. Therefore, for a moving target P, the visual motion with nonzero Δx (i.e., the visual motion in the x direction) will identify itself from the static background in the general case, which implies that the motion of the target in the x direction is not zero (i.e., $S_x \neq 0$). In other words, the correspondence pair of such a point will violate the epipolar line constraint for static points (i.e. $\Delta x = 0$). Note that this represents the general cases of independent moving targets.

(4) *3D constraints.* Even if the motion of the target happens to be in the direction of the camera's motion (i.e., the y direction), we can still discriminate the moving target by examining 3D anomalies. Typically, a moving target (a vehicle or a human) moves on a flat ground surface (i.e., road) over the time period during which it is observed through the leading and trailing edges of video images with a limited field of view. We can usually assume that the moving target shares the same depth as its surroundings, given that the distance of the camera from the ground is much larger than the height of the target. A moving target in the direction of camera movement, when treated as a static target, will show 3D anomaly - either hanging up above the road (when it moves to the opposite direction, i.e., $S_y < 0$), or hiding below the road (when it moves in the same direction, i.e., $S_y > 0$). Note this is only the special case of independent moving targets.

After a moving target has been identified, the motion parameters of the moving target can be estimated. We first estimate the depth of its surroundings and apply this depth Z to the target, then calculate the object motion s_y using Eq. (3-5), and (S_x, S_y) using Eq. (3-6), knowing the visual motion (Δx, Δy) measured in the stereo mosaics.

3.2 REAL-WORLD ISSUES AND MULTI-VIEW MOSAICS

In real applications, there are three sets of challenging problems. These include camera motion estimation in practical cases, mosaic generation with more general camera motion, and occlusion and stereo matching issues in a pair of stereo mosaics. For some issues, we will give very brief discussions

and point to related work. More details will be given for dynamic stereo mosaic generation, and multi-view pushbroom mosaics for dealing with occlusions, stereo matching and moving target detection.

3.2.1 Camera orientation estimation

The first problem is that the camera usually cannot be controlled with ideal 1D translation and camera poses are unknown; therefore, camera orientation estimation (i.e., dynamic calibration) is needed., external orientation instruments, i.e., GPS, INS and a laser profiler, are used in an aerial video application to ease the problem of camera orientation estimation (Zhu, et al, 2004; Zhu et al 2005). More general approaches using bundle adjustment techniques (Triggs, et al, 2000) are under investigation for estimating camera poses of long image sequences, which is one of the challenging issues of our stereo mosaic approach, and of video sequence analysis in general. Here we focus on other technical issues of the problem, and use an ideal 1D camera translational model to show the principle of the dynamic pushbroom stereo mosaics, without loss of generality. In our experimental analysis, we either assume that the extrinsic and intrinsic camera parameters are known at each camera location, as in theoretical analysis, or use a simplified version of camera orientation estimation, in which only four camera parameters are used. The four parameters are translation components in the X and Y directions, a heading angle, and a scaling factor. An underlying assumption in the practical treatments is that, (1) if the translational component in the Z direction is much smaller than the distance itself, we use a constant scaling factor in the interframe motion estimation and image rectification for each frame to compensate for the Z translation; and (2) the rolling and tilting angles are small so they are combined into the translations in the X and Y directions. The mosaics from real video sequences are generated from such camera orientation estimation model. We have found that 3D perception is compelling and 3D reconstruction results are reliable with such treatments, and the results could still be useful for image-based rendering and automated target detection.

3.2.2 Stereo mosaicing for dynamic scenes

The second problem is to generate dense parallel mosaics with a sparse, uneven, video sequence, under a more general motion, and for a complicated 3D scene. For the case of static scenes, a parallel ray interpolation method is proposed for stereo mosaics (PRISM) approach (Zhu, et al 2004) for generating a

generalized stereo mosaic representation for static scenes, under constrained 6 DOF motion. At the first look, the approach might not be applicable to dynamic scenes. But a carefully study shows that the PRISM approach designed for static scenes also works for dynamic scenes. Fig. 3-2 illustrates the basic idea of the PRISM algorithm in generating one forward-looking *dynamic* pushbroom mosaic (left mosaic with slit window location d_{yl}). In the figure, (T_{x1}, T_{y1}, T_{z1}) and (T_{x2}, T_{y2}, T_{z2}) denote two consecutive camera locations, at time t_1 and t_2, respectively. From each of the two frames, only one scan line (the fixed line) can be directly used for the mosaic since it is generated from the correct viewing direction. For any other point P between these two fixed lines, its parallel-perspective projection needs to be interpolated from its matching pair in the two frames, (x_1, y_1) and (x_2, y_2), respectively. If the point P is a static point, the triangulation gives its correct 3D location $P(X,Y,Z)$, and its back-projection gives the necessary parallel view as seen from the "interpolated" camera location (T_{xi}, T_{yi}, T_{zi}), where

$$T_{yi} = T_{y1} + \frac{y_1 - d_{yl}}{y_1 - y_2}(T_{y2} - T_{y1}), T_{xi} = T_{xl}, T_{zi} = T_{zl} \tag{3-7}$$

assuming $T_{x1} = T_{x2}$ and $T_{z1} = T_{z2}$ under the ideal 1D camera motion case. However, for a moving point (from 3D positions P_{t1} to P_{t2}), the triangulation does not give us its right 3D coordinates, but the back-projection will create an image of the moving point P_{ti} that should be seen at the "interpolated" time t_i , i.e. at camera location (T_{xi} , T_{yi} , T_{zi}), which is a linear interpolation between time t_1 and t_2. This naturally gives a linearly pushbroom scanning of the moving point. Under the linear motion assumption, the mosaic coordinates of the pair of point are

$$y_i = \frac{F}{H}T_{yi} + d_{y1}, x_i = x_1 \tag{3-8}$$

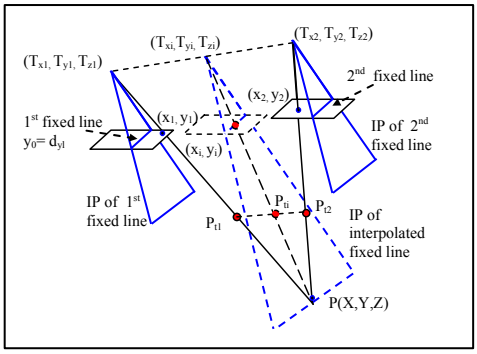


Figure 3-2 Ray interpolation for a dynamic scene

This is an important finding since the mosaicing algorithms developed for static scenes can be directly applied to dynamic scenes.

In principle, the PRISM approach needs to match all the points between the two overlapping slices of the successive frames to generate a complete parallel-perspective mosaic. In an effort to reduce the computational complexity, a fast PRISM algorithm has been designed (Zhu, et al 2004), based on the proposed PRISM method. It only requires matches between a set of control point pairs in two successive images, and the rest of the points are generated by warping a set of triangulated regions defined by the control points in each of the two images. The proposed fast PRISM algorithm can be easily extended to use more feature points (thus smaller triangles) in the overlapping slices so that each triangle really covers a planar patch or a patch that is visually indistinguishable from a planar patch, or to perform pixel-wise dense matches to achieve true parallel-perspective (pushbroom) geometry.

3.2.3 Multi-view pushbroom mosaics for dynamic scenes

Finally, 3D reconstruction and motion detection from two widely separated stereo mosaics raise challenging issues. A pair of stereo mosaics (generated from the leading and trailing edges) is a very efficient representation for both 3D structures and target movements. However, there are two remaining issues. First, stereo matching will be difficult due to the largely separated parallel views of the stereo pair, resulting in large perspective distortions and varying occlusions. Second, for some unusual target movements, e.g. moving too fast, changing speed or direction, we may either have two rather different images in the two mosaics (if changing speed), or we see the object only once (if changing direction), or we never see the object (if it maintains the same speed as the camera and thus never shows up in the second edge window).

Therefore, we propose to generate multi-view mosaics (more than 2), each of them with a set of parallel rays whose viewing direction d_{yk} is between the leading and the trailing edges, d_{y0} and d_{yK}, respectively (Fig. 3-3, k = 0, 1, …, K). The multiple mosaic representation is still efficient. Moreover, there are three benefits of using them. First, multiple pushbroom mosaics can be used for image-based rendering with stereo viewing in which the translation across the area is simply a shift of a pair of mosaics, and the change of viewing directions is simply a switch between two consecutive pairs of mosaics. Second, it

eases the stereo correspondence problem in the same way as the multi-baseline stereo (Okutomi & Kanade, 1993), particularly for more accurate 3D estimation and occlusion handling. In the stack of pushbroom mosaics, different sides of a 3D object will be represented in mosaics with various viewing angles. Each of these mosaics with parallel projections views the scene from a unique parallel viewing direction, thus captures surfaces of 3D objects visible from that direction (refer to Fig. 3-9 a-c for three views of pushbroom mosaics with different sides of buildings visible in different mosaics). In Section 3.3, we will discuss in details a new method to extract both of 3D structures and moving targets from multiple dynamic pushbroom mosaics. We will also discuss the possibility to extract and represent occluding regions in Section 3.4.2.

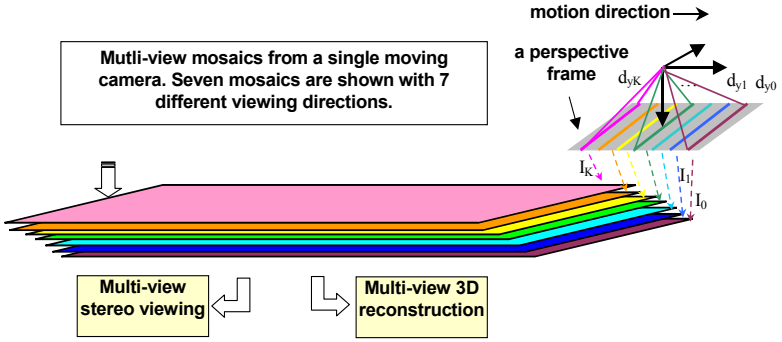


Figure 3-3 Multi-view pushbroom mosaics

Third, multiple mosaics can also facilitate 3D estimation of moving targets, and increase the possibility to detect moving targets with unusual movements and also to distinguish the movements of the specified targets (e.g., ground vehicles) from those of trees or flags in wind. Here we want to briefly discuss how multi-view mosaics can be used to estimate 3D structure of a moving target on the ground. In order to estimate the height of a moving target from the ground, we will need to see both the bottom and the top of an object. A pair of pushbroom mosaics with one forward-looking view and the other backward-looking view exhibits obvious different occlusions; in particular, the bottom of a target (e.g., a vehicle in Fig. 3-4a) can only be seen in one of the two views. However, any two of the multi-view pushbroom mosaics, if both with forward-looking (or backward-looking) parallel rays, will have almost the same occlusion relation to satisfy the condition for height estimation.

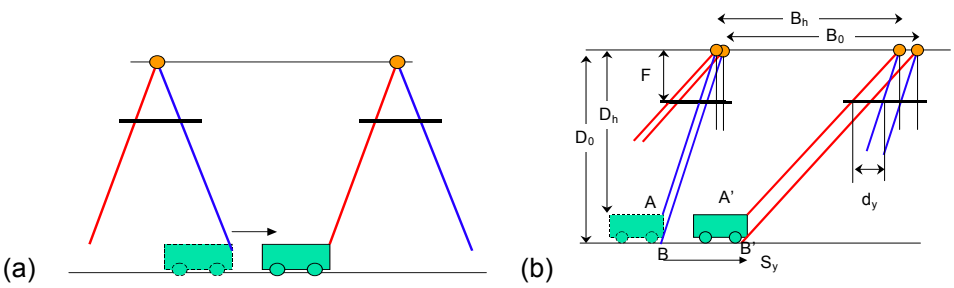


Figure 3-4 Height from dynamic pushbroom stereo: (a) an infeasible pair; (b) a feasible pair

Fig. 3-4b illustrates the case of a pair of backward-looking pushbroom stereo mosaics. Point *A* and *B* are two points on a target (vehicle), one on the top and the other on the bottom. Both of them are first seen in the mosaic with parallel rays of a smaller oblique angle, and then seen in the mosaic with parallel rays of a larger oblique angle. The distance between the two different rays within an image frame is still defined as d_y. The visual motion in the y direction is Δy_h and Δy_0, respectively, and can be measured in the stereo pair. Between the two parallel views, let us assume the motion of the target is S_y in 3D space and s_y in the mosaic images. Then the depths of the points on the top and on the bottom are

$$Z_h = F\frac{B_h - S_y}{d_y} = H(\frac{d_y + \Delta y_h - s_y}{d_y}) \tag{3-9}$$

and

$$Z_0 = F\frac{B_0 - S_y}{d_y} = H(\frac{d_y + \Delta y_0 - s_y}{d_y}) \tag{3-10}$$

respectively. Depth Z_0 of the bottom point could be obtained from the surroundings (ground) of the target. Then, the object motion s_y (and therefore S_y) can be calculated using Eq. (3-10). Finally, the depth of the point on the top, Z_h, can be estimated using Eq. (3-9), given the known visual motion of that point, Δy_h, and its independent motion component s_y obtained from the bottom point *B*.

3.3 3D and Motion Content Extraction

Using the advantageous properties of multi-view mosaics, we propose a unified approach to perform both stereo matching and motion detection. In a set of pushbroom mosaics, I_0, I_1, …, I_K, generated from a

video sequence, at slit window locations d_{y0}, d_{y1}, …, d_{yK} (see Fig. 3-4), the leftmost mosaic I_0 at the location d_{y0} is used as the reference view, therefore color segmentation is performed on this mosaic, and the so called *natural matching primitives* (explained below) are extracted. Multiple natural matching primitives are defined with each homogeneous color image patch, which approximately corresponds to a planar patch in 3D. The representations are effective for both static and moving targets in man-made urban scenes with objects of largely textureless regions and sharp depth boundaries. Then matches of those natural matching primitives are searched in the rest of the mosaics, one by one. After matching each stereo pair, a plane is fitted for each patch, and its planar parameters are estimated. Then, multi-view matches are performed, and therefore multiple sets of parametric estimates for this planar patch are obtained. The best set is selected as the final result by comparing match evaluation scores. Local and global spatial constraints are also explored to improve the robustness of the 3D estimation. The moving targets are detected after the "3D alignments" of the scene.

The multi-view dynamic stereo mosaic approach has the following four stages: (1) natural patch-based stereo matching; (2) plane estimation from multiple views; (3) plane merging and updating using local and global scene constraints; and (4) moving object extraction using the dynamic pushbroom stereo geometry. We will describe the approach in detail in the following subsections.

3.3.1 Patch-based stereo matching

Stereo matching is applied first on a pair of stereo mosaics. Let the leftmost (i.e., reference) mosaic and the second mosaic be denoted as I_0 and I_1, respectively. First, the reference mosaic I_0 is segmented into homogeneous color image patches. In our current implementation, the mean-shift-based approach (Comanicu & Meer, 2002) is used; but other segmentation methods can also be used for this purpose. In practice, over-segmentation (into small patches) is undertaken for ensuring homogeneity of each patch to enable accurate 3D recovery; however, a segmentation with larger patches will result in higher compression ratio of the video sequence.

The segmented image consists of image regions (patches), $\{\mathbf{R}_i, i = 1, \ldots, N\}$, each with a homogeneous color $\mathbf{c}_i$ and is assumed to be a planar region in 3D space. All the neighboring patches, $\{\mathbf{R}_{ij}, j = 1, \ldots, J\}$, are also recorded for each patch $\mathbf{R}_i$, The boundary of each patch, $\mathbf{b}i$, is extracted as a closed curve. Then

we use a line fitting approach to extract feature points for stereo matching. The boundary of each patch is first fitted with connected straight-line segments using an iterative curve splitting method. The connecting points between line segments are defined as *interest points*, $\mathbf{p}_{il}$, = 1, …, L, around which the natural matching primitives are defined.

For each interest point, the best match between the reference and target mosaics is searched within a preset search range. Instead of using the conventional window-based match, we define the so-called *natural matching primitives* (Fig. 3-5) to conduct a sub-pixel stereo match. Note that the natural matching primitives around the detected interest points, instead of line segments or the patches, are the features to be matched. We define a region mask W_l of size w×w centered at each interest point $\mathbf{p}_{il} = (x,y) \in \mathbf{R}_i$, such that

$$W_l(u,v) = \begin{cases} 1, if\ (x+u, y+v) \in \mathbf{R}_i \\ 0, otherwise \end{cases} \tag{3-11}$$

The size w of the mask is adaptively changed depending on the actual size of the region $\mathbf{R}_i$. In order that a few more pixels (1-2) around the region boundary (but not belonging to the region) are also included so that we have sufficient salient image features to match, a dilation operation is applied to the mask W_l to generate a region mask covering pixels across the depth boundary. Fig. 3-5 shows four such windows for the four interest points for the top region of the box. Note the yellow-shaded portions within each rectangular window, i.e., the natural matching primitives, indicating that the pixels for stereo matching cover the depth boundaries. They are called “natural matching primitives”, because these primitives define the natural structures of the salient visual features, in terms of sizes, shapes and locations. Each natural matching primitive in the reference image is defined by its location (x,y) on the patch’s boundary $\mathbf{b}_i$, and the pixels belonging to the patch, which is represented by the size of a rectangular window and the mask (together they form a “natural” window as a yellow region in Fig 6). To this point, the attributes of each region (patch) $\mathbf{R}_i$ can be summarized as:

$$\mathbf{R}_i = (\mathbf{c}_i, \mathbf{b}_i, \{\mathbf{R}_{ij}, j = 1,...,J\}, \{\mathbf{p}_{il}, W_l, l = 1,...,L\}), i = 1,...,N \tag{3-12}$$

which includes its color, boundary, J neighboring regions, L interest points and the corresponding masks.

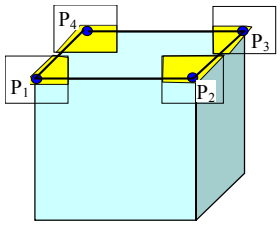


Figure 3-5 Natural matching primitives

The weighted cross-correlation, based on the natural window centered at the interest point *(x, y)* in the reference mosaic, is defined as

$$C(\Delta x, \Delta y) = \frac{\sum_{u,v} W_l(u,v) I_0(x+u, y+v) I_1(x+u+\Delta x, y+v+\Delta y)}{\sum_{u,v} W_l(u,v)} \tag{3-13}$$

Note that we still carry out correlation between two color images but only for those interest points on each region boundary, and for each interest point, the calculation is only carried out on those pixels within the region and on the boundaries. A sub-pixel search is performed in order to improve the accuracy of 3D reconstruction; and a match is marked as *reliable* if it passes the crosscheck (e.g., as in Scharstein & Szeliski, 2002), i.e. the matches from the reference to the target and from the target to the reference are consistent. For the simplicity of representation, we still use Eq. (3-11) to represent the region $\mathbf{R}_i$, with a note that the number (L) of reliable interest points used in the following steps may be smaller than the total number of interest points.

The matching process consists of the following two steps.

Step 1: *local match.* For each interest point, in order to find a reliable corresponding point, the natural matching strategy is carried out with a multi-scale approach, in that the search ranges and search steps are changed adaptively (from large to small). First, the natural matching strategy is applied to each interest point $\mathbf{p}_{il}$ (l=1...,L) of a region (patch) $\mathbf{R}_i$ (i=1,…, N) in the reference I_0, within preset (large) search range (S_h, S_v) in both the horizontal (y) and vertical (x) directions, and a preset (large) search step *s*. Note that the pushbroom stereo geometry produces image displacement in the y direction, but to account for camera calibration and orientation estimation error, a search within a much smaller range in the x

direction is also performed. If a reliable match is obtained, and a new set of parameters (S_h, S_v and s) are calculated based on the first run (i.e., the search range is narrowed to neighborhood of corresponding point with a finer step, therefore S_h, S_v and s are reduced). Then, the natural matching is applied again, with the updated parameters. The same procedure is carried out recursively until convergence, i.e., s become a fraction (therefore match results are sub-pixel accurate). Usually the match procedure converges in three iteration steps.

Step 2: *Surface fitting.* Assuming that each homogeneous color region $\mathbf{R}_i$ is planar in 3D, then a 3D plane can be generated as

$$a_iX+b_iY+c_iZ=d_i \tag{3-14}$$

which is represented in the camera coordinate system as shown in Fig. 3-2, is fitted to each region after obtaining the 3D coordinates of the interest points of the region using the pushbroom stereo geometry (Eqs. 3-1 and 3-2).

We use a standard RANSAC method (e.g., Medioni & Kang, 2004) to fit planes. In our implementation, a plane is fitted by randomly selecting three reliable interest points, and then using the plane parameters, all reliable interest points are warped from the reference view onto the target view. For each reliable interest point, the distance between the warped interest point and its corresponding target point (from local match) is calculated, and if the distance is less than 1 pixel, the point is claimed to be the one that supports the fitted plane. The total number of supports is denoted as C, and the RANSAC process stops if C/L is larger than 65%, where L is the total number of reliable interest points. The number of the random selections of three points is set to N_{max} = 50. In other words, the RANSAC process will stop either at 50 iterations or when the number of the supporting reliable points exceeds 65% of total reliable points. Then the best set of the plane parameters is selected as the initial 3D estimation of the planar patch. In the latter case, the region is marked as a *reliable* patch (in 3D estimation), therefore a unreliable patch at this point is the one whose number of reliable interest points is smaller than 3, or the total number of planar supports does not exceed the required percentage (i.e. 65% in our experiments). In the end, there are three categories of patches: those with reliable plane estimation under the plane fitting criterion (C_i=2),

those with unreliable plane estimation (C_i=1), and those without any plane estimation (C_i=0). At this point, each patch's representation can be updated as

$$\mathbf{R}_i = (\mathbf{c}_i, \mathbf{b}_i, \{\mathbf{R}_{ij}, j = 1,...,J\}, \{\mathbf{p}_{il}, W_l, l = 1,...,L\}, C_i = 0,1,\text{or } 2, \boldsymbol{\Theta}_i = (a_i, b_i, c_i, d_i)), i = 1,...,N \quad (3\text{-}15)$$

The plane parameter set $\boldsymbol{\Theta}_i$ exists if $C_i \neq 0$. All the patches will go to the next stage for further processing.

Before we go to the next stage, we want to summarize the advantages of the patched-based natural matching primitives for stereo matching. First, treated separately, natural matching primitives on a patch represent the most salient visual features of the patch, and only contain pixels on that patch. Therefore, more accurate matches can be found for the patch that is textureless within and has a sharp depth boundary around. Second, taken together, more accurate and more robust results can be expected since these natural matching primitives are fitted on a single planar surface. Finally the algorithm is very efficient since only interest points of a region are matched in order to obtain the 3D of all the points within the region.

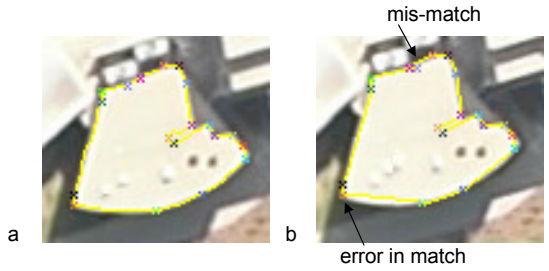


Figure 3-6 An example of region matching results. The matches are marked as "X", with corresponding colors.

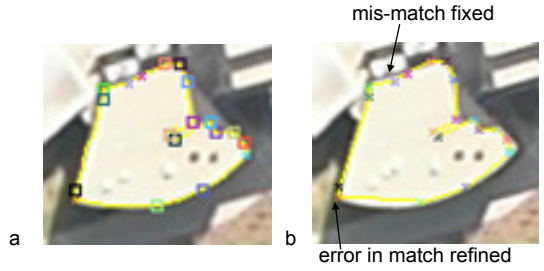


Figure 3-7 An example of surface fitting results. Both the mismatch and the small error in the initial match are fixed

Fig. 3-6 shows a real example of a natural-window-based stereo matching result for a static object (the roof of a building). The 19 interest points that are detected and their correspondences are marked on the boundaries in the left and right images, respectively. One mismatch and a small error in match are also indicated on the images. Fig. 3-7 shows the results of fitting and back-projection of the fitted region onto the right image. The 15 seed interest points (out of 19) used for planar fitting are indicated on the left image as squares. Both the mismatch and the small error in the initial match are fixed.

3.3.2 Refining plane parameters with multiple mosaics

After the above stereo matching is applied to the first pair of stereo mosaics, I_0 and I_1, initial estimations of the 3D structure of all the patches (regions) in the reference mosaic are obtained. Further matches between the reference mosaic I_0 and each of the rest of the mosaics, I_2, …, I_K, are then conducted. The initial visual displacement of each interest point on a patch is predicted from the result of this point estimated from the first stereo pair. From Eq. (3-2), we know the visual displacement Δy is proportional to the selected "disparity" (d_y) for a pair of stereo mosaics for any static point, i.e.,

$$\Delta y = (\frac{Z}{H} - 1) d_y \tag{3-16}$$

Therefore, the visual displacement of the interest point in consideration can be predicted except when the point is on a moving object, which will be reconsidered in the moving target detection stage. Assume that the visual displacement for an interest point is Δy_1 between I_0 and I_1, where $d_y = dy_0 - dy_1$, then between I_0 and I_k, where $d_y = dy_0 - dy_k$, the predicted visual displacement is

$$\Delta y_k = (\frac{d_{y0} - d_{yk}}{d_{y0} - d_{y1}}) \Delta y_1 \tag{3-17}$$

For refining the initial estimates of visual displacements, the two-step algorithm in Section 3.3.1 is modified to obtain new plane parameters for each pair of stereo mosaics, with a very good initial estimation to start with to reduce the search range.

From the K pairs of stereo mosaics, up to K sets of plane parameters $\Theta_{ik} = (a_{ik}, b_{ik}, c_{ik}, d_{ik}), k=1,\ldots,K$, are obtained for each region (patch) in the reference mosaic (some regions have fewer than K sets of

available plane parameters due to the lack of sufficient numbers of interest points, or unreliable plane fitting). In order to obtain the most accurate plane parameters for each planar patch, the following steps are performed. First, for each pair of stereo mosaics, the patches in the reference mosaic are warped to the target mosaic in order to compute a color sum of square differences (SSD) for each region, between warped and original target images. Generalizing Eq. (3-1) to K views, and with 3D planar parameter estimation, we have

$$\begin{cases} x_k = F\dfrac{X}{Z} \\ y_k = F\dfrac{Y}{H} - (\dfrac{Z}{H} - 1)d_{yk} \\ a_k X + b_k Y + c_k Z = d_k \end{cases} \tag{3-18}$$

where the subscript i is dropped for simplifying the notations. Given a point $\boldsymbol{p}\ (x_0,y_0) \in \mathbf{R}_i$ in the reference view I_0,.

$$\mathbf{p}_k = \Psi_k(\mathbf{p}) \tag{3-19}$$

Then the color SSD of the k^{th} interest point of the region $\mathbf{R}_i$ can be calculated as

$$SSD_{ik} = \sum_{\mathbf{p}\in\mathbf{R}_i} |\mathbf{I}_k(\Psi_k(\mathbf{p})) - \mathbf{I}_0(\mathbf{p})|^2, k = 1,2,...,K \tag{3-20}$$

where $\mathbf{I}_0$ and $\mathbf{I}_k$ are the color vectors in the reference and the k^{th} target views. Then, among all the estimates for each patch, the set of plane parameters with the least SSD value is selected as the best plane estimate. With multi-view refinements, the plane parameters and their categories in Eq. (3-15) are updated; some regions under the categories C_i = 0 or 1 may be upgraded into the category C_i = 2 under both the plane fitting criterion and multi-view refinement.

Note that using the knowledge of plane structure (i.e., 3D orientation), the best angle to view the region can be estimated, where the viewing direction of the selected mosaic (among all the possible viewing directions) is the closest to the plane norm direction. For example, for the side of a building that faces the right (refer to Fig. 3-2), the best match could be obtained from the first pair of stereo mosaics. If the view angle is equal to or greater than 90 degrees (relative to the plane norm), then the region will not be visible.

Incorporating this information, the SSD calculations are only carried out for those patches between the reference and target mosaics if the plane norms have less than 90-degree view angles from the viewing directions of the mosaics. Experimental results of improvements in 3D reconstruction will be shown in Section 3.5 with both real and simulated video sequences.

3.3.3 Plane updating using neighbors and global scene constraints

After the plane parameters with the smallest SSD value have been obtained for each region $\mathbf{R}_i$, we will have a close look at the best SSD of each region within category $C_i = 2$, under both the plane fitting criterion and multi-view refinement. If the SSD value is larger than a preset threshold T_i, then the patch is moved to *unreliable* category ($C_i = 1$) under plane fitting, multi-view refinement and SSD evaluation, therefore the attributes in Eq. (3-15) are further updated. Note that the SSD of the region $\mathbf{R}_i$ is calculated as the sum of all the pixels of 3 color components in the region, therefore T_i is defined as

$$T_i = Q_i \times 3 \times D^2 \qquad (3\text{-}21)$$

where Q_i is the total number of pixels in the region $\mathbf{R}_i$, and D is the threshold of the difference between two corresponding components. In our experiments, we set D = 16 pixel levels of 512 possible differences. We have found that quite some small regions around a large region corresponding to a surface (or part) of a 3D object are generated by color segmentation, and are either marked as unreliable or without plane estimation. Therefore, we use two methods to update the plane parameter estimations: neighbor patch supporting and global scene constraints.

In the neighborhood supporting strategy, we perform a modified version of the neighboring plane parameter hypothesis algorithm (Tao, et al, 2001) to infer better plane estimates. Based on our region categorization, the main modification is that the parameters of a neighboring region are adopted only if it is marked reliable and the best neighboring plane parameters are accepted only when the match evaluation cost (SSD) using the parameters is less than the threshold T_i for the i^{th} region $\mathbf{R}_i$. Our neighbor supporting algorithm has the following steps.

(1). Select reliable regions $\{\mathbf{R}_{i,j1}, \mathbf{R}_{i,j2},\ldots, \mathbf{R}_{i,jM}\}$ from the set of neighboring regions $\{\mathbf{R}_{ij}, j = 1,2,\ldots J\}$ for the current region $\mathbf{R}_i$, including the current region, therefore M<= J+1.

(2). Apply the parameter set Θ_{jm} (m=1, 2,…,M) to the region $\mathbf{R}_i$, to calculate the corresponding $SSD_{i,jm}$(m=1, 2, …,M), using Eq. (3-20).

(3). Select the parameter set Θ_{jm}(1 <= m <= M) that gives the smallest SSD, for the current region.

With the neighborhood supporting, a un-estimated (C_i = 0) or un-reliable region (C_i = 1) can be upgraded to a reliable region (with C_i = 2) if its best SSD is smaller than the threshold T_i; the plane parameters of most of the regions can be refined no matter what categories they initially were. Further, if the neighboring regions sharing the same plane parameters, then they are then merged into one reliable region. This step is performed recursively till no more merges occur. We prefer to have false negatives than false positives, and the former will be handled in the next stage – moving object detection.

We have also explored global scene constraints to improve the robustness of 3D reconstruction for highly cluttered urban scenes, where a lot of small patches are generated. In a typical urban scene, many surfaces such as facades, rooftops, roads, etc., share the same plane directions. Therefore, in applying the global scene constraints, after an initial pass of plane parameter estimation with multiple views, the top several dominant plane directions are obtained by a simple clustering algorithm on those reliable regions. Then the following two steps are performed.

(1) For those regions that either are marked as unreliable (due to plane fitting or SSD evaluation), or do not obtain sufficient good local matches (L<3), the parameters of the dominant planes can be used to guide the search and the refinement of their matching and plane fitting steps. Since each plane only has 4 parameters (a, b, c and d), and the norm of each dominant plane provide 3 of them (i.e., a, b and c), the rest of the job is simply to compute the variable d. Therefore, for each region with at least one reliable local match among the detected interest points, we plug this reliable match into the plane equation using each of these domination plane norms, to obtain possible estimations of d. Then, we compute the SSD of the corresponding patch pair (i.e. the warped reference patch and the original target patch) based on each estimate of the parameter d, and finally select the one with the smallest SSD score as the result.

(2) After applying the global scene constraints, neighborhood hypothesis (as discussed above) is applied to *all* the regions to generate more reliable and accurate 3D estimation results.

Experimental results on plane merging and local/global scene constraints will be shown in Section 3.5, with both simulated and real video sequences.

3.3.4 Moving object detection

After the plane merging stage, most of the small regions are merged together and marked as reliable. Moving object patches that move along epipolar lines should also obtain reliable matches after the plane merging step, but they appear to be “floating” in air or below the surrounding ground, with depth discontinuities all around it. In other words, they can be identified by checking their 3D anomalies (Section 3.2, observation (4)). This is mostly true for aerial video sequences, where ground vehicles and humans move on the ground.

In general cases, most of the moving targets are not exactly on the direction of the camera’s motion, therefore, those regions should have been marked as unreliable in the previous steps. Regions with unreliable matches fall into the following two categories: (1) moving objects with motion not obeying the pushbroom epipolar geometry; (2) occluded or partially occluded regions, or regions with large illumination changes. For regions in the second category, their SSDs in stereo matching evaluation are always very high. The regions in the first category correspond to those moving objects that do not move in the direction of camera motion; therefore they do not obey the pushbroom stereo epipolar geometry. Therefore, for each of these regions, we perform a 2D-range search within its neighborhood area. If a good match (i.e., with a small SSD) is found within the 2D search range, then the region is marked as a *moving* object. We can also take advantage of the known road directions, to more effectively and more reliably search for matches of those moving vehicles. The road directions can be derived from 3D reconstruction results, e.g., in a city scene, the norm directions of the two dominant planes of the building façades surrounding the ground area on which the moving objects reside.

In the current implementation of moving target detection (ground vehicles) from aerial images, large occluded regions are still not well processed and consequently confuse the moving target detection as described above. Therefore, the size of each region is also taken into account to classify it as a moving target. Only if the region size is less than 300 pixels, it goes through the moving target detection procedure.

The moving target detection steps are summarized as follows.

(1) For all reliable regions with less than 300 pixels, the 3D anomaly condition is checked. If one of the following conditions is satisfied, then a region $\mathbf{R}_i$ goes through 2D region search to find its motion parameters (S_x, S_y), and is marked as a moving target if the SSD is smaller than the preset threshold T_i: (a) the height of the region $\mathbf{R}_i$ is 20 meters higher than the average height of the neighboring regions $\{\mathbf{R}_{ij}\}$; or (2) the height of the region $\mathbf{R}_i$ is 10 meters lower than the average height of the neighboring regions.

(2) For *all* unreliable regions with less than 300 pixels, the epipolar constraint is applied. Each region $\mathbf{R}_i$ in this class goes through 2D neighborhood search to find its motion parameters (S_x, S_y), and is marked as a moving target if the SSD is smaller than the preset threshold T_i.

At the end of all the four stages, a region $\mathbf{R}_i$ is represented as the following form:

$$\mathbf{R}_i = (\mathbf{c}_i, \mathbf{b}_i, \{\mathbf{R}_{ij}, j = 1,...,J_i\}, C_i, \mathbf{\Theta}_i = (a_i, b_i, c_i, d_i), \mathbf{m}_i = (S_{xi}, S_{yi})), i = 1,..., N \tag{3-22}$$

where C_i is redefined as reliable static region (C_i = 2), moving target (C_i=1), and unreliable region (C_i=0), $\mathbf{m}_i$ is the motion vector if the region is a moving target. Note that we have removed the interest points and natural matching primitives from each region in Eq. (3-22), which are only used during the 3D estimation process. And more precisely, the number of neighboring region for the region $\mathbf{R}_i$ is noted as J_i (i=0,…, N).

3.4 CB3M: CONTENT-BASED 3D MOSAICS

The output of the two-phase processing – pushbroom mosaicing and content extraction, is a content-based 3D mosaic (CB3M) representation. It is a highly compressed visual representation for very long video sequences of a dynamic 3D scene. In the CB3M representation, the panoramic mosaics are segmented into planar regions, which are the primitives for content representation. Each region is represented by its mean color, region boundary, plane normal/ distance, and motion direction/speed if it is a dynamic object. Relations of each region with its neighbors are also built for further object representations (such as buildings, road networks) and automatic target recognition.

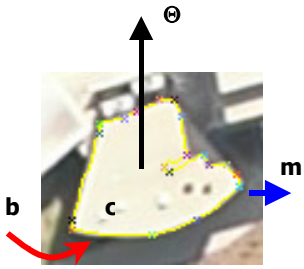


Figure 3-8 Content-based 3D mosaic representation

3.4.1 Basic content-based 3D mosaic representation

In our current basic implementation, a content-based 3D mosaic (CB3M) representation is a set of video object (VO) primitives (i.e., patches, e.g. in Fig. 3-8) that are defined as

$$\mathbf{CB3M} = \{\mathbf{R}_i, i = 1, \ldots, N\} \quad (3\text{-}23)$$

where Ri is defined in Eq. (3-22). As a summary, they are explained below:

(1) N is the number of VOs, i.e., "homogeneous" color patches (regions);

(2) $\mathbf{c}_i$ is the color (3 bytes) of the i^{th} region;

(3) $\mathbf{b}_i$ is the 2D boundary of the i^{th} region in the left mosaic, chain-coded as $\mathbf{b}_i = \{(x_0, y_0), G_i, b_1, b_2, \ldots b_{Gi}\}$, where the starting point (x_0,y_0) has 4 bytes, and each chain code has 3 bits. G_i is the number of boundary points (which needs 4 bytes each) and $G = \sum G_i$ is the total for all regions;

(4) $\{\mathbf{R}_{ij}, j = 1,\ldots, J_i\}$ is the list of the labels of neighboring regions of the ith region, each needs 4 bytes (assuming on average the number of neighboring regions for each region is J, i.e. $J = (1/N) \sum J_i$);

(5) $C_i > 0$, if the region is a static patch with reliable plane parameter and the value is the average match cost (0-255) of all pixels of the region over three channels; $C_i = -1$ if the region is a moving target (therefore with mi, see (7)); $C_i = 0$, if the region does not obtain reliable 3D estimate (unreliable, maybe occluded regions or moving objects).

(6) $C_i = 2$, if the region is a static patch with reliable plane parameters (see (6)); $C_i = 1$, if the region is a moving target (therefore with mi, see (7)); $C_i = 0$, otherwise (unreliable, maybe occluded regions).

(7) $\mathbf{\Theta_i} = (a_i, b_i, c_i, d_i)$ represents the plane parameters of the region in 3D, 4 bytes for each parameter; and

(8) $\mathbf{m_i}$ represents the M motion parameters of the region if in motion (e.g. M =2 for 2D translation (S_x, S_y) on the ground).

Therefore the total data amount is (without counting C_i)

$$N_{color} + N_{boundary} + N_{neighbor} + N_{structure} + N_{motion}$$

$$= 3N + (8N+3G/8) + 4JN + 4*4N+4M*N_m$$

$$= (27+4J)N+3G/8+4MN_m \text{ (bytes)} \qquad (3\text{-}24)$$

when each of the motion and structure parameters needs 4 bytes. In the above equation, N_m is the number of moving regions (which is much smaller than the total region number N). Note that the VO primitives are those patches before region merging in order to preserve the color information.

The proposed CB3M representations are highly compressed visual representations for very long video sequences of dynamic 3D scenes. The representations could fit into the MPEG-4 standard (Koenen, et al, 1997), in which a scene is described as a composition of several Video Objects (VOs), encoded separately.

The CB3M construction and representation provides the following benefits for many applications, such as urban transportation planning, aerial surveillance, robot navigation and urban modeling. A long image sequence of a scene from a fly-through or drive-through is transformed in near real time into a few large FOV *panoramic mosaics*. This provides a synopsis of the scene with all the 3D objects and dynamic objects in a single view. The *3D contents* of the CB3M representation provide three-dimensional measurements of objects in the scene. Since each object (e.g. a building) has been represented into 3D planar regions and their relations, further object recognition and higher-level feature extraction are made possible. The *motion contents* of the CB3M representation provide dynamic measurements of moving targets in the scene. For example, in traffic monitoring, the motion and 3D contents not only provide information about the vehicles' directions and speeds, but also the traffic situation of a road segment since each road "region" can also be extracted based on its 3D information and shape, and the statistics of the moving objects on the road can provide very useful traffic information. Finally, the CB3M

representation is *highly compressed*. Usually a compression ratio of thousands to ten thousands can be achieved. This saves space when a lot of data for a large area need to be archived

3.4.2 Discussions: occlusion representation and higher level object representation

Since the basic CB3M representation is a set of planar patches with shape and appearance properties, it can be naturally extended to represent relations between regions, and occluded regions that are not visible or only partially visible in a single reference mosaic used as the base image of the basic CB3M representation. In the current implementation, only 3D parametric information of planar patches in the reference mosaic is obtained. Since different visibilities are shown in mosaics with different viewing directions, we want to extend the approach presented in Section 3.3 to produce multiple depth maps with multiple reference mosaics and then integrate the results by performing occlusion analysis. The neighboring regions of each patch have been extracted in the patch and interest point extraction step. This lays a solid foundation for object recognition and occlusion handling, which will be our future work. Then an extended content-based 3D mosaic representation can be generated by inserting the occluded regions in the basic CB3M representation, similar to the layered representation we have proposed in Zhu and Hanson (2004). In the end, the extended CB3M representation will have the following three components:

(1) A base layer that consists of a set of planar patches corresponding to the reference mosaic;

(2) A set of occluded patches that are not visible in the reference mosaic, but are visible in other views, together with the corresponding viewing direction information for these patches; and

(3) All the neighboring regions of each patch, including the base patches and occluded patches.

With these three components, and the corresponding viewing direction information, the extended content-based 3D mosaic representation can be easily converted into other representations, such as digital elevation map, and be used for image-based rendering since both the shape/appearance information and the viewing information are available. Furthermore, developing higher-level representations that group the lower-level natural patches into objects (vehicles, buildings, roads, humans) are also possible, for applications such as automated target recognition and 3D model indexing.

3.5 EXPERIMENTAL RESULTS AND ANALYSIS

The proposed approach for the content-based 3D mosaic representations was applied to multi-view pushbroom mosaics generated from real world video sequences. Here we present two examples: the flyover of a campus scene and the flyover of a New York City (NYC) scene. We also performed evaluations on the accuracy of 3D and motion estimation and the compression power of the CB3M representation on a simulated video sequence generated with the ground truth data, which is presented in Appendix A. The analysis on computation time in both stereo mosaicing and content extraction is provided in the Appendix B.

3.5.1 Results and analysis on a simulated scene

Nine parallel-perspective stereo mosaics were generated from a simulated video sequence of a simulated scene with ground truth data of both 3D and moving targets (Fig. 3-9). The sequence was generated using the following parameters. The virtual “aircraft” with a video camera flew at a 300-meter height above the ground along a 1D translational direction, and the motion direction is perpendicular to the optical axis of the camera. The focal length of the camera is 3000 pixels (as in Eqs. (3-1), (3-4) and (3-6)), and the camera moves with a constant speed. The 3D “buildings” are with heights from 5 to 120 meters above the ground, with different roof shapes (rectangular, round, frontal, ridged, slanting, and/or with small attachments). There are occlusions between buildings. Each of the eight moving objects has a height from 2 to 5 meters, and undertakes a 2D translational motion with constant velocity during the period of the capture of the total 1640 frames of images, except the one labeled as “1” in Fig 10a, which varies in velocity. The velocity of the motion of each moving target is represented in centimeter (cm) per frame. Nine 1-column width slit windows are used to generate the nine mosaics (refer to Fig. 3-3), every pair of the two consecutive windows has a 40-pixel distance, and hence the total distance between the first and the last slit windows is 320 pixels. Fig. 3-9 only shows three of the nine mosaics, (a) the leftmost, (b) the center, and (c) the rightmost views. Varying occlusions/visibilities can be seen in these mosaics. The change of velocity of the 1st moving target can be seen from the varying sizes of its images in the three mosaics. From the nine mosaics, we use the leftmost mosaic as the reference image to match with the other eight mosaics. For each region in the reference mosaic, there are 8 plane estimation results,

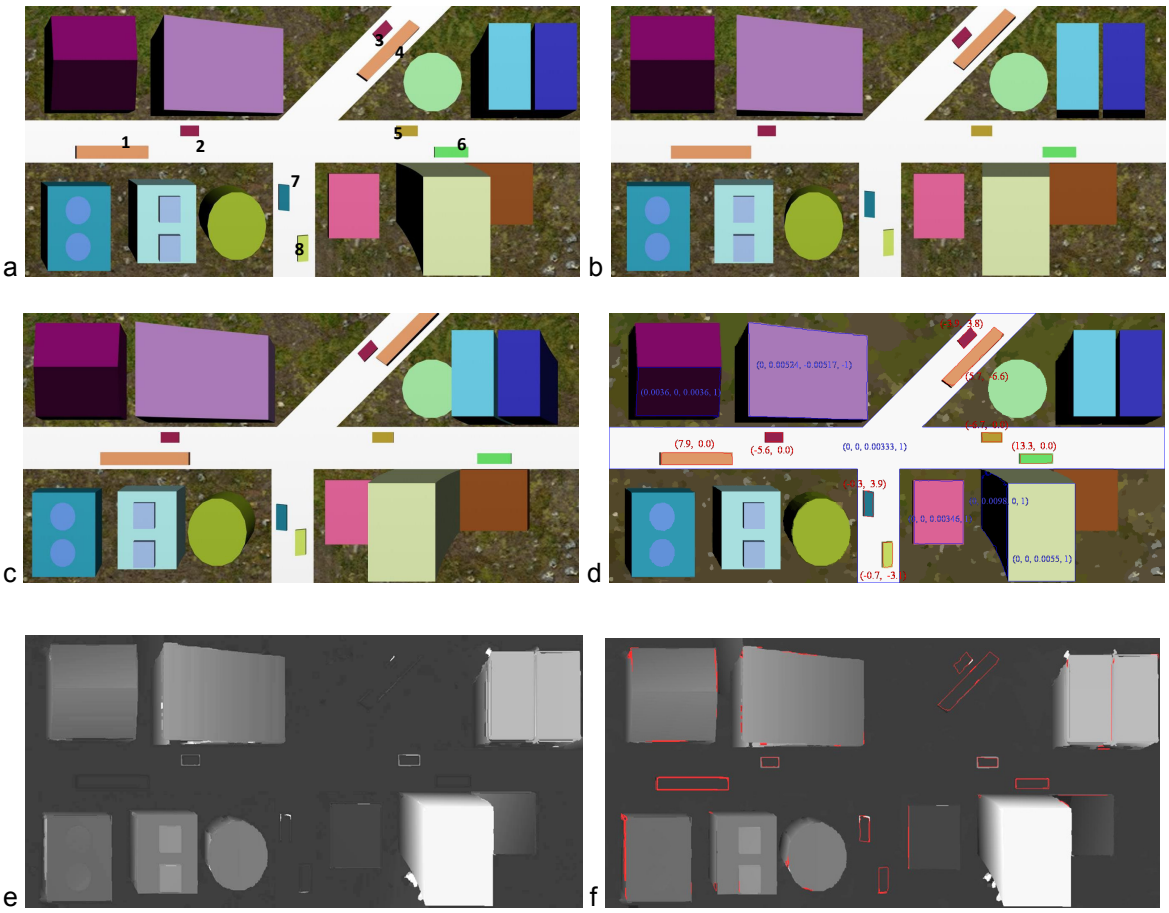


Figure 3-9 (a) The leftmost, (b) center and (c) rightmost views of the nine mosaics of a simulated scene. The final CB3M representation is shown in (d). Each region is rendered by its average color. Plane parameters *(a,b,c,d)* (in blue) and boundaries for several representative surfaces, and motion displacements (s_x, s_y) (in red) of the detected moving targets are labeled in (d). For comparison, (e) and (f) show the rendered "height" maps of the scene from the stereo matching results from the 1st stereo pair only, and from all the mosaics, respectively. Finer and more accurate results are obtained in (f). Regions marked in red are the "outliers" that will be passed to the moving target test; some of them are due to occlusions at depth boundaries rather than independent motion, but they are too thin or too small to be a real moving targets. The detected moving targets are shown in (d).

and the best estimate is selected for the 3D parametric representation of the region. The final "height" map (Fig. 3-9f) is rendered as a map of heights of objects from the ground, i.e. $-H\Delta y/d_y$, (normalized to a range from 0 to 255 for display). For comparison, we have also generated a height map (Fig. 3-9e) from

the stereo matching results of only the first and the second mosaics (without region merging). It can be seen that by using the best parameter selections from multi-view mosaics and utilizing the plane merging step, finer 3D results are obtained for many building roofs, and more accurate results are obtained for sides of buildings.

3.5.2 Results on real video data: a campus scene

The first real video sequence we tested our approach on is for a campus scene captured by a camera on a light airplane flying about 300 meters above the ground. The camera was calibrated using some ground truth data. The image resolution is 640*480. Nine mosaics were generated from the 1000-frame aerial video. Fig. 3-10a shows a pair of stereo mosaics (embedded in red and green-blue channels, respectively) from the nine mosaics, and two close-up windows are marked in the stereo mosaics, which include both various 3D structures and moving objects (vehicles). Fig. 3-10b is the “height” map (corresponding to the reference mosaic) using the proposed method. Fig. 3-10c and Fig. 3-11d, Fig. 3-10e and Fig. 3-11f show the images of the two close-up windows and the corresponding “height” maps. Note the sharp depth boundaries are obtained for the buildings with different heights and various roof shapes. The average heights of the buildings marked as A, B, C, D and E in Fig. 3-10d and Fig. 3-10f are 11.5m, 5.8m, 5.4m, 14.9m and 7.8m, respectively. The long building (D) has a slanting roof (left side is higher). Even though we have not conducted an accurate evaluation due to the lack of ground truth data, these estimations are consistent with the real heights of these buildings. The moving objects that have been detected across all the nine mosaics are shown by their boundaries (in red). Those vehicles that are not detected by our algorithm are marked by rectangular bounding boxes; they are either stationary (as those in the boxes 2 and 3), or deformed differently across the mosaics due to the changes of motion in velocities (as in the box 1) and directions (as in the box 4).

Figure 3-10. 3D and motion from multi-view stereo mosaics of an aerial video sequence. (a) A pair of stereo mosaics from the total nine mosaics;'' (b) height map of entire mosaic; (c) close-up of the 1st window marked in (a); and (d) the height map of the objects inside that window, with the detected moving targets marked by their boundaries and those not detected by rectangular boxes; (d) close-up of the 2nd window marked in (a); and (f) the height map of that window.

The CB3M mosaic (of the first window in Fig. 3-10a) is shown in Fig. 3-11, with a color, a boundary, plane parameters and a motion vector (if in motion) for each patch (region). Again we examine the compression of the real video sequence from two steps: stereo mosaicing and then content extraction. For the real image sequence, we have 1000 frames of 640*480 color images, so the data amount is 879 MB. The size of pair of the stereo mosaics (Fig. 3-10a) is 4448*1616*2, which has 41MB (without compression and with more than half empty space due to the fact that the mosaics go in a diagonal direction). The two mosaics in high-quality JPEG format only have 2*560 KB; therefore, a compression ratio of about 800 is achieved

for the stereo mosaics (the first step). If all the nine mosaics are saved for mosaic-based rendering, then the data amount is 9*560KB so the compression ratio will be 179.

Then after color segmentation, 3D planar fitting and motion estimation, we obtained the CB3M representation of the video sequence, with the total number of the natural regions N = 6,112 and the total number of boundary points G = 420,445. The total amount of data in its CB3M representation is 316 KB (with a header). This real file size is consistent with the estimation of data amount using Eq. (3-24), which is about 315 KB. The data amount is reduced to 90 KB with a simple lossless Winzip on the CB3M data; therefore, the compression ratio is about **10,001**. Note that the CB3M representation in Fig. 3-11 consists of regions corresponding to rather large object surfaces in order to rapidly obtain robust 3D structures. However, fine details are not preserved. In our previous experiments, we over-segmented the reference mosaic so that finer details of the scene can be coded. In that case the compression ratio was still over 2000.

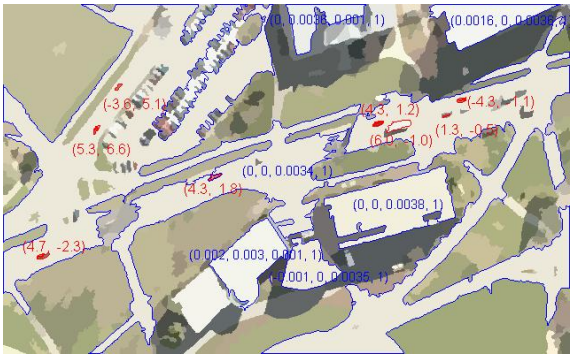


Figure 3-11. Content-based 3D mosaic representation of an aerial video sequence. Only a window is shown, with some of the regions labeled by their boundaries and plane parameters (in blue), and the detected moving targets marked by their boundaries and motion vectors (in red).

3.5.3 Results on real video data: a NYC scene

The NYC mosaics were generated from a video sequence from a NYC HD (high-definition) aerial video dataset (vol. 2) we ordered from http://www.artbeats.com/prod/browse.php. The video clip, NYC125H2, has about 25 seconds, or 758 frames of high-definition progressive video (1080*2000). Rooftops and city streets are seen as the camera looks ahead and down in a close flight just over One Penn Plaza and

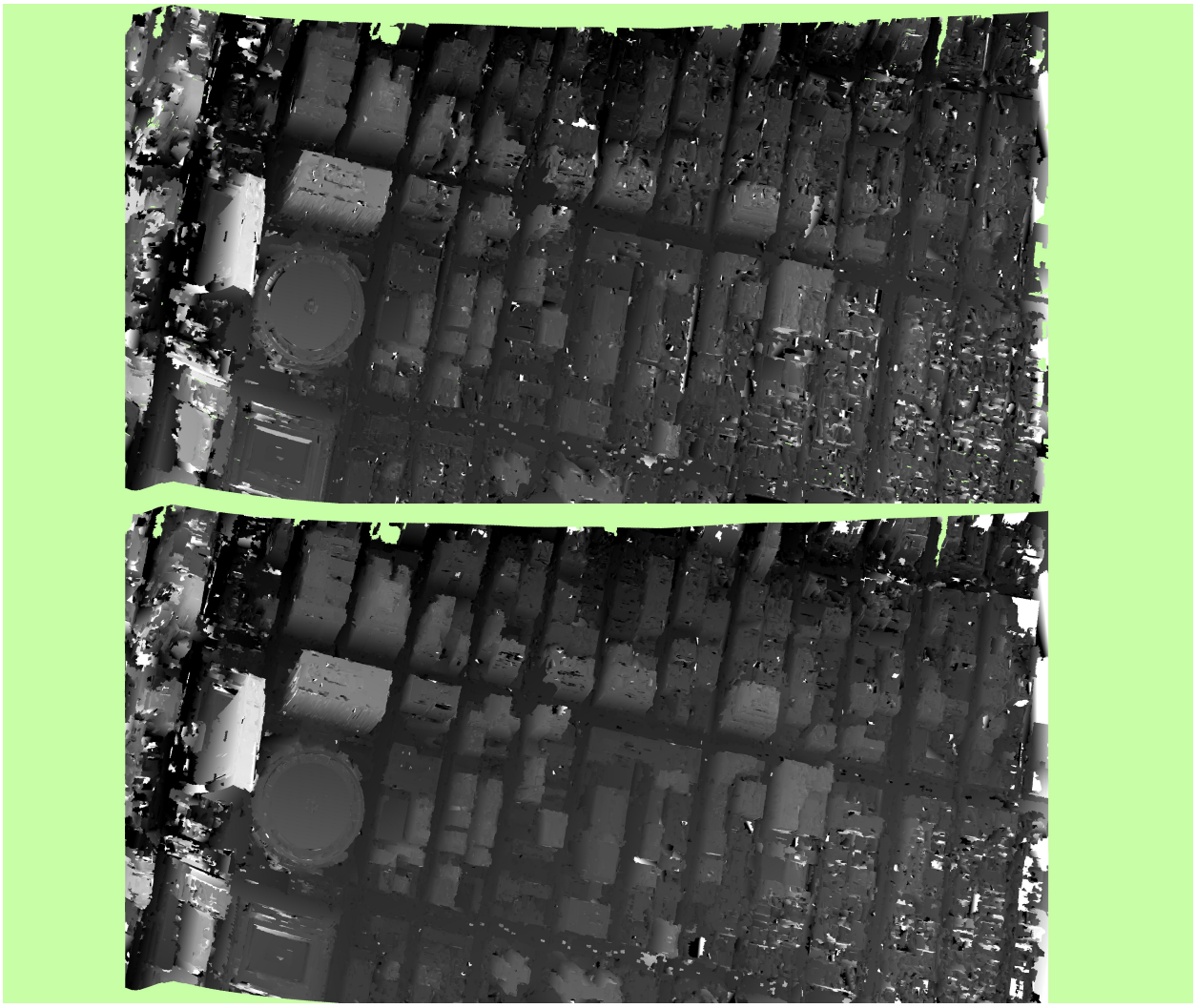
beyond in New York City. Yellow taxicabs make up a noticeable percentage of the vehicles traveling the grid of streets in this district of mostly lower-rising buildings, but have a few high-rise buildings. You may view the low-resolution version of the video following the link we have provided above. Our main task is to recover the full 3D model of the area automatically, with cluttered buildings with various heights, from less than ten to more than a hundred meters. Fig. 3-12 shows one of the four multi-view mosaics generated and used for 3D reconstruction and moving target detection. The mosaic that is shown here has been turned 90 degrees, therefore the camera moves in the direction from the left to the right in the mosaic. The size of the mosaic is 4816 (W) x 2016 (H). The camera slightly tilted to the up-right side so the ground plane in the mosaic is not leveled. You can clearly see this effect in the depth maps in Fig. 3-13.

This data set is very challenging due to the cluttered buildings and complex micro-surface structures that produce a lot of small homogeneous color patches after color segmentation. The regions with low-rising buildings (the right-hand side of the mosaic) do not have salient visual features and sufficient disparity for reliable depth estimation. So in this example, we also applied the Manhattan world geometric constraint (Coughlan & Yuille, 1999) to further refine the 3D reconstruction results. As shown in Fig. 3-12, most of the planes (roads, rooftops and facades of buildings) are either perpendicular or parallel to each other; therefore, they consist of three orthogonal domination plane directions. In our experiments, among of all regions that have successfully obtained plane-fitting results from multi-view mosaics, those with reliable matches are used to automatically vote for the three domination planes. The three plane norms are [5.544, 1.360, 1.000], [-0.792, 3.837, 1.000] and [-0.026, -0.318, 1.000]. A simply cross-product check verifies they are almost orthogonal to each other (The angles between them are 85.52^{o}, 86.03^{o} and 92.69^{o}). The information of these three domination plane directions is very useful in both refining the 3D reconstruction and extracting moving targets. For this, the two-step strategy in using the global scene constraints discussed in Section 3.3.3 is applied.

Then, the rest of regions, i.e., the “outliers”, go through the moving object detection test. We use the method as presented in chapter 3.3.4, and for this NYC data, we take advantage of the known road directions, to more effectively and more reliably search for matches of those moving vehicles. The road directions are derived from the two dominant planes of the building façades (the third one is for the ground and rooftop).

Figure 3-12. A 4816 (W) x 2016 (H) mosaic from a 758-frame high-resolution NYC video sequence. The Manhattan world geometric constraint is illustrated on the mosaic

Fig. 3-13 shows the 3D reconstruction results of the NYC video data, all represented in the leftmost mosaic - the reference view. In Fig. 3-13a, the height map is rendered from the 3D structure result reconstructed from the first pair of stereo mosaics. It can be seen that the right- hand side has many spurious small regions. Fig. 3-13b shows the height map rendered from the result from the integration of the 3 stereo pairs of the four mosaics. It is obvious that the height map has improved significantly. The height map looks much smoother; many spurious depth estimations and small regions without reliable estimations are filled. Fig. 3-13c shows the colored coded height map from multi-view mosaics (same as Fig. 3-13b). The color bar on the right-hand side shows the correspondences of colors and height values. Due to the lack of the flight and the camera parameters, we roughly estimate the main parameters of the camera (i.e., the height H and the focal length F) from some known buildings. However, this gives us a good indication of how well we can obtain the 3D structure of this very complex scene. For example,



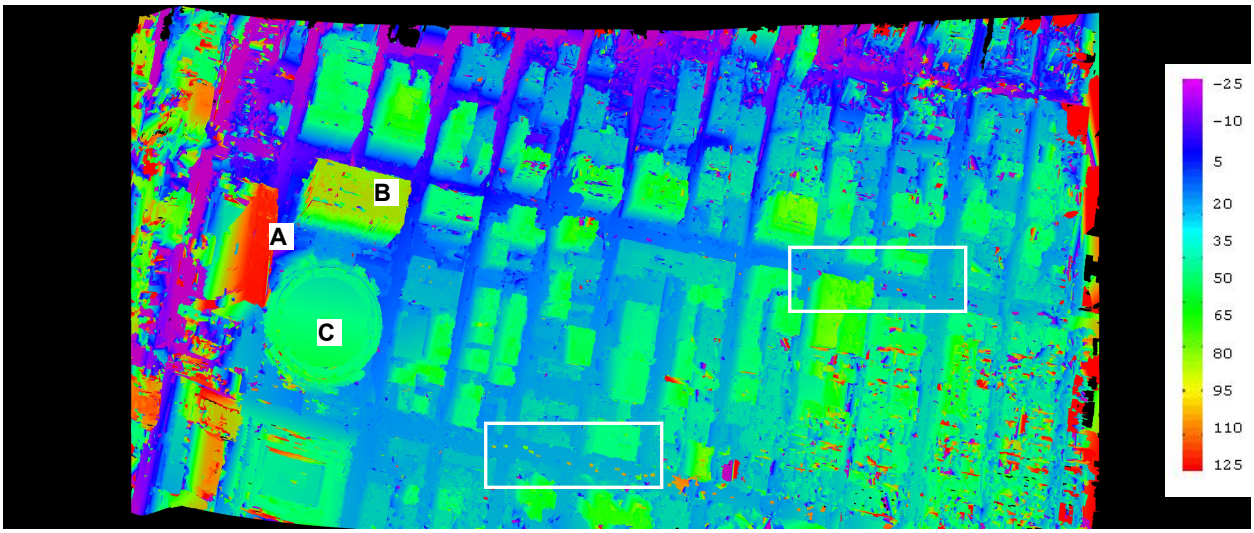


Figure 3-13. (a) Depth from a pair of mosaics, (b) from four mosaics, and (c) color-coded depth map of (b)

the average heights of the three buildings at One Penn Plaza (marked as A, B and C in Fig. 3-13c) are 105.32 m, 48.83 m, and 19.93 m, respectively. Our approach handles scenes with dramatically varying depths. Readers may visually check the heights of those buildings with GoogleEarth. Note that the camera was not pointing perpendicularly down to the ground and therefore the reconstructed ground is tilted. This can be seen from the colors of the ground plane.

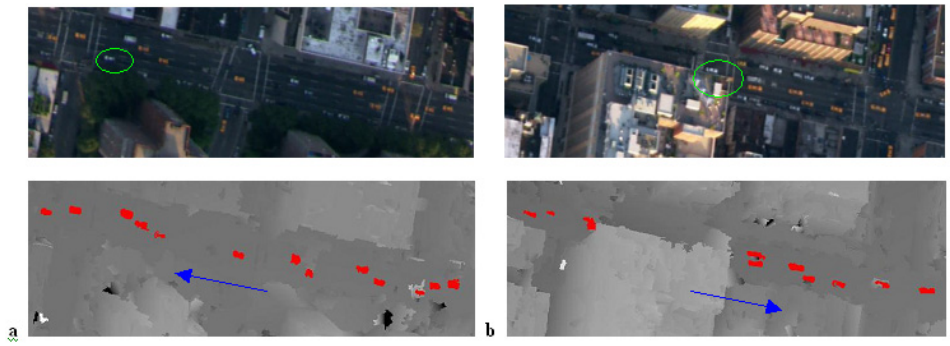


Figure 3-14. Moving target detection using the road direction constraint. In the figure (a) and (b) are the corresponding color images and height maps of the 1st (bottom-left) and 2nd (top-right) windows in Fig. 3-12, with the detected moving targets painted in red. The two circles show the three moving targets that are not detected. The arrows indicate the directions of the roads along which the moving targets are searched.

The moving objects (vehicles) create “outliers” in the height map, as can be clearly seen on the color-coded map. For example, on the one-way road indicated in the first window in Fig. 3-12, vehicles moved from the right to the left in the figure, therefore, their color-encoded “height values have more red/yellow colors (i.e., the estimated heights are much higher than the ground if assumed static). On the other hand, on the one-way road indicated in the second window in Fig. 3-12, vehicles moved from the left to the right in the figure, therefore, their color-encoded “height values have more blue colors (i.e., the estimated heights are much lower than the ground if assumed static). After further applying the constraint of road directions that we obtained from the dominant plane voting, moving targets are searched and extracted. In Fig. 3-14, all of the *moving* targets (vehicles) are extracted, except the three circled in the figure. These three vehicles are merged with the road in color segmentation. Other vehicles that are not detected were

stationary; most of them are on the orthogonal roads with red traffic signals on for stop, and a few parked on these two one-way roads.

Chapter 4 SCENE UNDERSTANDING FROM CONTENT BASED 3D MOSAICS

The content based 3D mosaics, corresponding to phase first and second of Fig 1.1, is discussed in Chapter 3. In the third phase, the CB3M representation is used for a higher-level scene understanding. The proposed graph-based method parses the CB3M in three steps that is the generation of three layers, including a surface layer, a structural layer and a cluster layer. The first layer (*surface layer*) is generated by merging neighboring patches that approximately share the same physical planar surface. The merging procedure is a graph labeling problem solved by a graph searching approach. Starting from a node (patch) labeled as the first *surface*, its neighboring patch will be labeled (merged) into the same group if they share similar colors, plane equations and are connected in space.

The second layer (*structure layer*) will be generated by grouping related surfaces into a structure. For example, the roof, facades of a building shall be grouped into one. The grouping will be based on color and prior knowledge of a structure.

The third layer (*cluster layer*) means to label those small, noisy patches into clusters. In other words, a surface or a structure is too small either in size or does not have reliable 3D estimate should be clustered with its neighbor with similar color and closeness in 3D (sharing longer boundary).

4.1 SURFACE LAYER GENERATION

In this layer, neighboring patches with similar physical and color properties are grouped together, using a graph search algorithm. The CB3M data structure is represented as a graph $G = (V, E)$.

Each patch is treated as a graph note, and a graph edge can be created if two patches are connected. A breadth first search is used to scan through all patches in the graph. A confidence measurement of 3D modeling is obtained by evaluating the different properties of patches, including shape and size of a patch, match cost in stereo and geometric smoothness among neighboring patches. The surface layer grouping is processed only on patches with enough confidence and in the order of patches following the rank of the confidence.

$$\mathrm{Conf}(\mathrm{i}) = \mathrm{C_a}(\mathrm{i})\,\mathrm{Conf_{shape}}(\mathrm{i})\,\mathrm{Conf_{match}}(\mathrm{i})\,\mathrm{Conf_{planesmooth}}(\mathrm{i}) \quad (4\text{-}1)$$

Where

$$\mathrm{Conf_{shape}}(\mathrm{i}) = 1 - \mathrm{e}^{-1/s^2} \quad (4\text{-}2)$$

$$\mathrm{Conf_{match}}(\mathrm{i}) = 1 - \mathrm{e}^{-M/m^2} \quad (4\text{-}3)$$

$$\mathrm{Conf_{planesmooth}}(\mathrm{i}) = \frac{\sum_j S_n(i,j) S_d(i,j)}{N}, \text{ } N \text{ is total number of patches} \quad (4\text{-}4)$$

Where *i* represents the *i*th patch and <i, j> are two neighborhood patches. The $\mathrm{Conf_{shape}}$ (Eq. 4-2) characterizes the confidence of plane fitting and matching in terms of the shape of patch, usually the plane fitting prefers convex iso-centeric shape, for example, a disk and square since the boundary of region distributed smoothly and has a better chance to get more accurate plane estimate. A variable $s=\text{eigenvalue}_1/\text{eigenvalue}_2$ where eigenvalue_1 and eigenvalue_2 are the larger and smaller eigenvalues of the boundary points of the patch fitted by an ellipse (therefore two eigenvalues in 2D space), respectively. The more the ratio *s* closes to 1, the more the boundary of the patch distributed smoothly.

The $\mathrm{Conf_{match}}$ (Eq. 4-3) encodes the relation between matching cost of warped patch, w.r.t. the estimated plane parameters. The smaller the match cost, the more likely the plane is estimated accurately. The variable m is the average match cost (0-255) of all pixels over three channels that is encoded in the CB3M representation. M is a constant (10 is used in the experiment).

The $\mathrm{Conf_{planesmooth}}$ (Eq. 4-4) indicates the confidence of the *i*th patch computed from the smoothness of plane estimate in a local neighborhood area, where *j* is index of the neighboring patches, and the geometric smoothness of two patches is measured by

$$\mathrm{S_n}(\mathrm{i},\mathrm{j}) = \mathrm{e}^{-\left(\frac{\mathrm{n_i}-\mathrm{n_j}}{\sigma_\mathrm{n}}\right)^2} \quad (4\text{-}5)$$

$$\mathrm{S}_d(\mathrm{i},\mathrm{j}) = 1 - \mathrm{e}^{-\left(\frac{\mathrm{n_i}\bar{\mathrm{x}}_\mathrm{j}+\mathrm{n_j}\bar{\mathrm{x}}_\mathrm{i}}{\sigma_\mathrm{d}}\right)^2} \quad (4\text{-}6)$$

While S_n denotes the smoothness of norm vectors (n_i and n_j) of two neighboring patches *i* and *j*, and S_d measures their distance smoothness (x_i and x_j) are coordinates of centroid of two patches so $n_i \bar{x}_j$ is the distance of the centroid of *i*th patch from the planar surface estimated from jth patch, while $n_j \bar{x}_i$ is the distance of the centroid of *j*th patch from the planar surface estimated from ith patch). σ_n and σ_d are constant.

Since the size of a patch might also affect the performance of plane estimation, the larger the patch is, the more robust the plane estimate. Hence the size of the patch is also taken into account to the estimate of confidence measurement and the parameter C_a in Eq. (4-1)

$$C_a(i) = 1 - e^{\frac{a_i}{c_1}} \tag{4-7}$$

Where a_i is the number of pixels in a patch and C_1 is a constant (we assume a patch with size greater than C_1 pixels has more chance to get good plane estimate).

Because the measurements C_a, $Conf_{shape}$, $Conf_{match}$ and $Conf_{planesmooth}$ are all in the range of [0 1], and Conf(i) is in the same range too. This confidence is used in the plane clustering step and the breadth first graph based search is performed by the rank of the confidence.

The similarity measurement of two neighbor planar patches (*similarity* (i, j) used in Table 1) is used to consider if two patches are clustered together in the surface layer, and it defined as

$$similarity(i,j) = S_n(i,j)\, S_d(i,j)\, S_c(i,j) \tag{4-8}$$

and

$$S_c(i,j) = e^{-\left(\frac{I_i - I_i}{\sigma_I}\right)^2} \tag{4-9}$$

While S_c measures the color difference of two neighboring patches and σ_I is a constant.

The complete work flow of the surface clustering is showed in Table 1. Note that patches without enough confidence values are not labeled (refer to Fig. 4-1 for the light blue boundaries).

The ground surface is segmented into many patches in the color segmentation step therefore remain the same in the CB3M representation (refer to the pink patch boundaries in Fig. 4-2). After the first layer – surface layer is built up, all patches shared similar geometry are labeled into single layer. Therefore, we obtain a new graph $Gs = (Vs, Es)$ – a surface layered graph, after merging, many neighbor nodes become a big node, so the edges among nodes are also modified and complexity of graph is much reduced .

$$E_{i,j} = \begin{cases} \text{collapsed}, & V_i \text{ and } V_j \text{ are merged into } G_s \\ kept, & \text{otherwise} \end{cases} \quad (4\text{-}10)$$

Table 1. Work flow of the generation of the surface layer

```
Sort patches by confidences in decreasing order => P;  (P(i): ith patch; P(k, j): the jth neighbor patch of the
kth patch)

Q = NULL; Labels = NULL; i=1;
For i=1…N          //N: the number of patches in image
  If Labels(P(i)) != Null or Conf(P(i)) < T_conf   //T_conf: a threshold of confidence
     Continue;
  End-if
  Enqueue(Q, P(i));
  While Q is not empty
      P(k) = Dequeue(Q);
      If Conf(P(k))<=T_conf
         Continue;
      End-if
      for j=1…M              //M: the number of the neighbors of P(k)
          if Conf(P(k,j)) >= Tc and Similarity(P(k), P(k,j)) < Ts
              Enqueue(Q, P(k,j));
              Labels(P(k,j)) = l;
           End-if
      End-for
  End-while
i++;
End-if
```

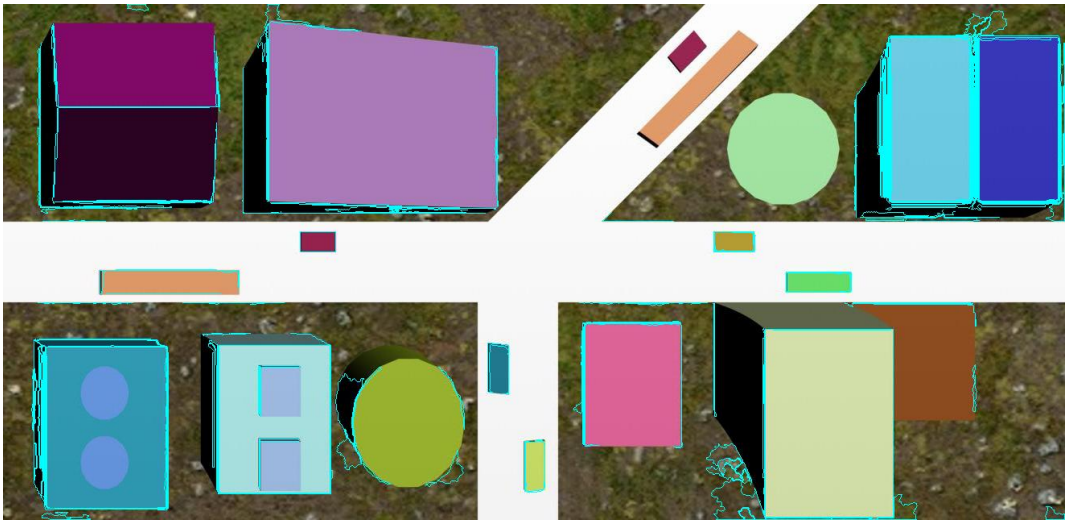


Figure 4-1. The original mosaic overlaid by patches without enough confidences (marked in light blue). Most of them are on the boundaries of buildings

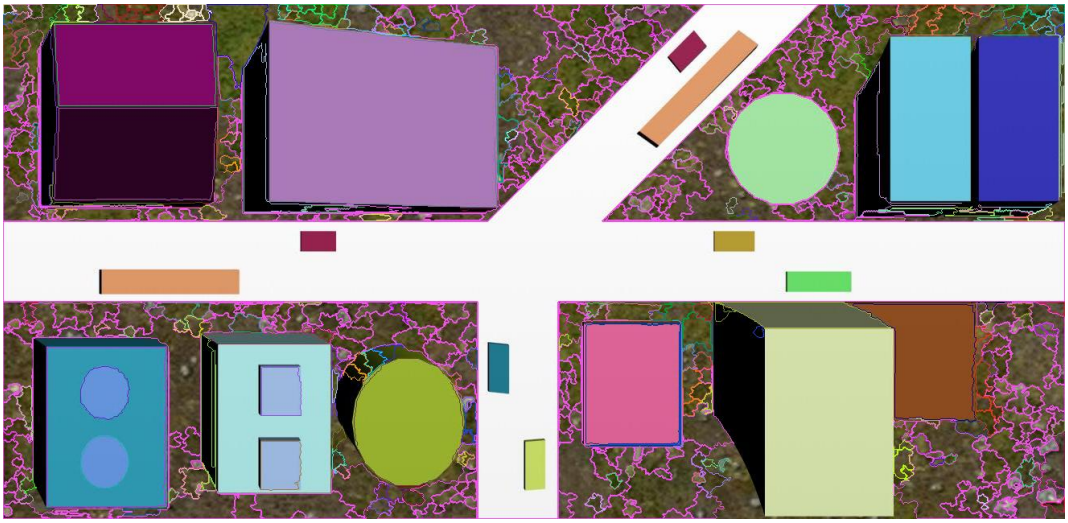


Figure 4-2. Surface layer generation. Patches shared with similar geometry are labeled into the same color on the boundaries. Most of patches on the ground plane are marked in same label (pink) after surface layer though a few of them are labeled in green, yellow and brown. All colors are randomly selected.

4.2 STRUCTURAL LAYER GENERATION

After this step, a structural layer is built and patches belong to one object, particular building in our experiment, are expected to be grouped together.

As a prior knowledge, assume we known the structure of buildings are mostly with box-shaped, consisting of one building top and four façades and we may represent a building by using a simple star graph model (Fig. 4-3), the center node (in orange) in the graph represents a building top, and a number of connected nodes indicate the four (and more) possible facades (they may not be all visible). Note that an edge in the graph represents not only a neighbor relation, but also has the geometric relations of two connected nodes, including orthogonality and other orientation relations. This simple graph model can roughly represent many buildings in the real scene. Although a building in upper left corner of the mosaic (Fig. 4-2) consists of two ridge roofs, this simple graph model can still be used after simply combining two rooftops together by checking the heights of these two surfaces. Since generating the structural layer is actually to label man-made buildings (including multiple surfaces) from the content-based 3D mosaic representation, after defining this simple graph model, the task becomes a subgraph matching problem (by looking for the star graph from our surface-layer graph, also called the subgraph-isomorphism problem). Even though this problem is a NP-complete problem, we can just apply a brute-force search method to find an approximated solution of the problem since our subgraph is a bit special and the center nodes has always distinguishable geometric properties, such as its height is significant higher than the rest of surfaces, particularly the ground plane surface. So in the first step, all surfaces with height greater than a threshold are labeled to be candidate building top nodes, the rest of tasks is just to confirm if at least a neighbor node is a façade node,

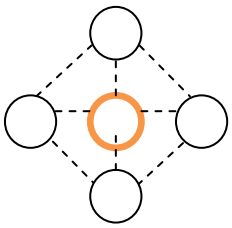


Figure 4-3. Star-shaped graph representation of building, the center node is building top and rest are façade, edges in the graph means two connected nodes are neighbor and perpendicular. The dashed lines represent the connections between two neighbor surfaces a not required.

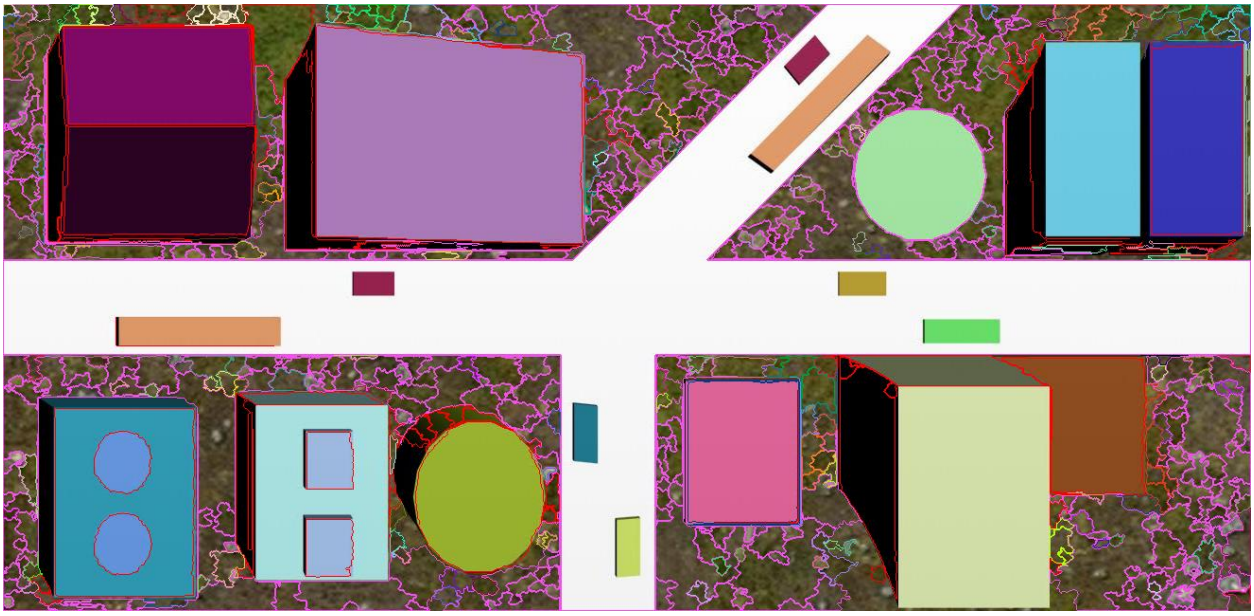


Figure 4-4, Structural layer generation. Patches on building are labeled in red. Note that how the roof and facades of e ach building is mostly labeled into one structure.

perpendicular to the center node. The whole process is fast since only a few surfaces have significant heights above the ground. Fig. 4-3 shows how the roof and facades of a building are grouped into a structure.

4.3 CLUSTER LAYER GENERATION

After man-made object - buildings are detected in the structural layer, many small patches, without enough confidence values due to inaccurate 3D modeling, can be clustered into neighbor patches. The patches with enough confidence are searched one by one using a breadth-first search similar to the one used in the surface layer generation. The similarity between two neighbor nodes is defined as

$$\text{similarity}_1(i,j) = S_c(i,j)\ \text{Conn}(i,j) \tag{4-11}$$

and

$$\text{Conn}(i,j) = \frac{lb_{i,j}}{lb_i} \tag{4-12}$$

While Conn(i,j) measures the likelihood that two neighbors are in same surface. lb_i is the total length of the boundary of the *i*th patch and $lb_{i,j}$ is the length of the boundary shared between the *i*th and jth patches. Under an assumption that any node must be grouped into one of its neighbors, any patches with mostly similar color and sharing longest boundary are likely in the same planar surface. Fig. 4-5 show the result after the generation of cluster layer, many yellow patches are clustered into their neighbor patches, so the structural layer (yellow bounded patches) becomes more completed.

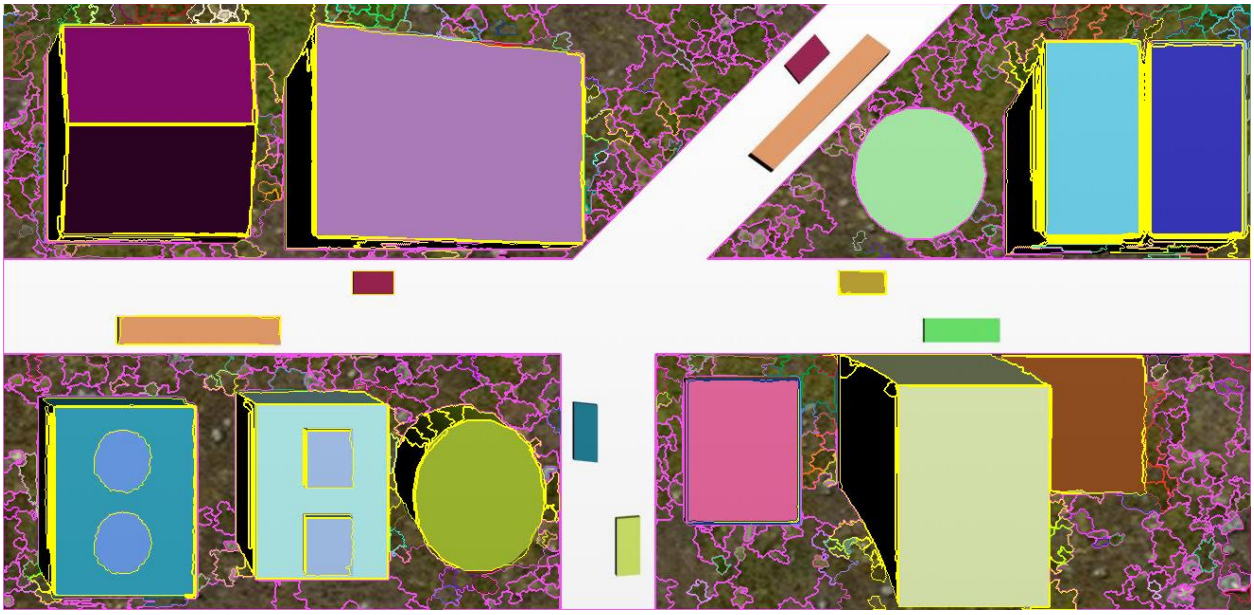


Figure 4-5. Cluster layer generation. Patches attached to buildings are labeled in yellow.

4.4 SUMMARY AND DISCUSSIONS

Based the output content-based 3D mosaic (CB3M) representation with a graph of planar patches, we parse a large-scale scene by exploring more higher-level constraints, including geometric, color and some prior knowledge of structure. The process is conducted hierarchically. Neighboring 3D patches are first grouped together into a larger patch with respective to their structures and colors, using a breadth first graph based search approach; then multiple neighboring and structural-related patches are further grouped based on their 3D relations. In the last step, the patches without enough confidences, therefore not involved into the first two layer generation, are merged into neighboring regions. This will serve as

the basis for our object detection and context understanding of large-scale 3D scenes, where roads, buildings, vegetation, and of course moving objects and their relations can be analyzed.

Chapter 5 3D MODELING FROM IMAGE SEQUENCE – USING PERSPECTIVE GEOMETRY

In the Chapter 3, we have introduced that a patch based 3D modeling algorithm to recover 3D model of a large-scale scene and produce a CB3M representation, using a pushbroom stereo mosaicing approach. In this chapter we will describe the adaptation of our efficient patch-based stereo matching algorithm to perspective stereo images captured by a stereovision head mounted on a mobile platform or carried by a blind person that moves with a more general motion trajectory. A subsampling needs to be conducted to reduce an original 2D/3D map from an original high resolution image to a low resolution sampling for stimulating visual implants (Second Sight, 2012), vibrotaciles (Palmer, et al, 2012) or tongue stimulation (Wicab, 2012). We will show that the patch-based method can generate meaningful segmentation results on both 2D and 3D maps; and then the most important information of the maps can be transduced to a blind user through certain types of alterative perception (visual implants, vibrotaciles or tongue stimulation). This holds promise that the above method could be used as visual prosthetics, so that blind or visually impaired users can have a better understanding of their surrounding environments.

In Section 5.1, the 3D modeling of 3D scenes using stereo images with perspective geometry is described. A smart sub-sampling method applied to the application of visual prosthesis will be discussed in Section 5.2. Some experimental results are provided in Sections 5.3. Finally a brief summary is given in Section 5.4.

5.1 3D MODELING FROM STEREO IMAGE

The method is adapted from the one we presented in Chapter 3, but for completion, we summarize it here. Here is an overview of our approach. The left camera of binocular stereo system serves as the reference camera. First, color segmentation is performed on the reference image, and the *natural matching primitives* (Fig. 3-5) are extracted. Multiple natural matching primitives are defined with each homogeneous color image patch corresponding to a planar patch in 3D space. The representations are effective for both outdoor and indoor scenes with objects of largely textureless regions and sharp depth boundaries. Then matches of those natural matching primitives are searched in the right image. After

matching the stereo pair, a plane is fitted for each patch, and a set of planar parameters for the planar patch is estimated.

In the following subsections, we will detail the three components of our approach: patch extraction, stereo matching, and plane merging & parameter refinement.

5.1.1 Patch and interest point extraction

First, the reference image of the stereo image pair is segmented, using the mean-shift based approach. The segmented image consists of image patches with homogeneous colors, and each of them is assumed to be a planar patch in 3D space. For each patch, its boundary is extracted as a closed curve. Then we use a line-fitting approach to extract feature points along the boundary for stereo matching: The boundary of each patch is first fitted with connected straight-line segments using an iterative curve splitting method. The connecting points (with large curvature) between line segments are defined as *interest points* around which the natural matching primitives are defined (Fig. 3-5).

Unlike outdoor scenes, many textureless surfaces in indoor scene may only contain a few interest points (for example, a rectangle may only have four interest points on the corners, see Fig. 3-5). Surface reconstruction may not be accurate if the plane fitting step only uses a small number of points. Fortunately, vertical lines yield more reliable matches between a pair of images that have horizontal epipolar lines as in the case of the ground stereovision head. Therefore, we pick up additional points on the boundary between two consecutive interest points when the line segment connecting them is non-horizontal. As a result, we will have more “interest points” on vertical lines. Now we are ready to perform the stereo match (Fig. 5.1).

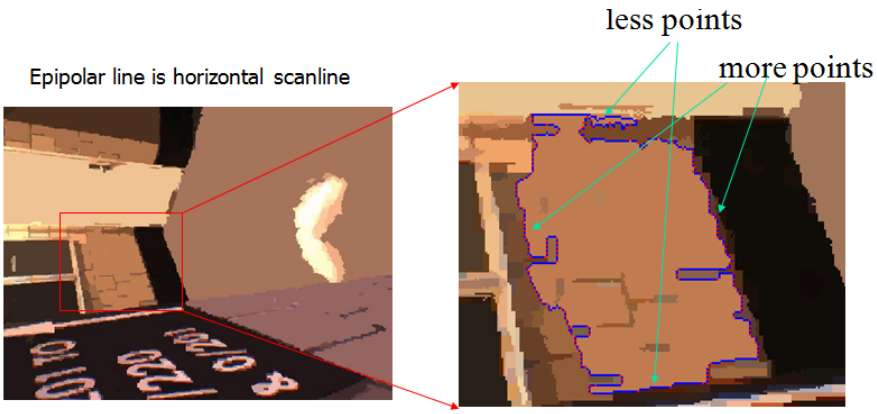


Figure 5-1. (a) The segmentation result of a reference image; (b) a close-up window shows more interest points are extracted in vertical boundary.

5.1.2 Stereo match

Let the left image and right image be denoted as I_1 and I_2, respectively.

Step 1: Local match. On any patch, for each interest point, the best match is searched within a preset search range.

A sub-pixel stereo match is conducted using the natural matching primitives (Sec. 3.3.1). We define a region mask M (natural window) of size *m*x*m*, centered at that interest point (Eq. 3-10).

The size m of the natural window is adaptively changed depending on the size of the region R. The SAD based on the natural window centered at the point (x, y) in the reference image, is defined as.

$$SAD(\Delta x) = \sum_{(x,y)\in M} | I_1(x,y) - I_2(x+\Delta x, y) | \tag{5-1}$$

Note that we still carry out sum of absolute difference calculation between two color images, but only on those interest points along each region boundary, and only with those pixels within the region and on the boundaries for each interest point. A confidence value *C* (0<C<=1) is defined as

$$C = \frac{SAD_1}{SAD_2} \tag{5-2}$$

where $SAD_1$and SAD_2 are the best (smallest) and the second best match score of each interest point. The smaller *C* is, the more confidence the match has. A sub-pixel search is performed in order to improve the accuracy of 3D reconstruction; and a match is marked as reliable if it passes a crosscheck followed (e.g., as in Scharstein & Szeliski, 2002).

Step 2: Surface fitting. Assuming that each homogeneous color region is planar in 3D, a 3D plane

$$aX + bY + cZ + d = 0 \tag{5-3}$$

which is represented in the camera coordinate system, is fitted to each region (patch) after obtaining the 3D coordinates of the reliable interest points of the region, since stereo camera head is calibrated (with

known intrinsic parameters). We use a robust RANSAC method to fit a plane. RANSAC randomly picks up samples, generates an estimation of the plane and votes from all samples (to either support or not support; each sample votes one ticket). The process is iterative and the estimating result obtaining the most tickets is selected as final estimation. In the voting step, interest points with higher confidences are able to vote more tickets, as

$$T = e^{1/C} \tag{5-4}$$

where T is the number of tickets obtained from one interest point, and C is the match cost defined in Eq. (5-2). Note that at this point the result of a patch after the above steps is in the form of a 3D planar equation and the boundary of each patch.

In general cases, the stereo matching described above is easy to recover a 3D model. However, in some applications, such as stereo vision for a climbing robot (Tang, et al 2009), or an application of robot "guide dog" as the visual aid for the navigation of blind people, cameras may be positioned with a small height above the ground plane. Therefore, the two views captured from a pair of stereo cameras are quite different and the disparities are large. On the other hand, if the stereovision head are mounted directly on the head of a blind person, the motion of the camera will be not well constrained. It increases the difficulty of stereo match. However, in both cases, cameras should move on a ground plane (or a rough planar surface) so the ground plane cue can be used to ease stereo match.

In the former case, assume the height of camera is known so the geometry of the ground plane in the camera coordinate is known. So the disparities of points on the ground plane can be estimated and verified using following preprocessing step.

Pre-step (using ground plane cue): a homography induce by the ground plane ($n^TX+d=0$) can be computed by

$$H = K(R - tn^T / d)K^{-1} \tag{5-5}$$

The stereo head is rectified and R is relative orientation (identity matrix), t is translation component (baseline) and K is the intrinsic matrix.

In a weak textureless scene, the single pixel based match couldn't work well. So we match a group of pixels on an image patch of the reference image towards the target image assuming image patch is a planar surface (the ground plane) in 3D. Therefore, for each region in the image I_1, the sum of absolute difference (SAD) is calculated between the warped (using homography) interest points of the reference image and the target image. The SAD is defined as

$$SAD = \sum_{x \in I_B} | I_1(x) - I_2(Hx) | \tag{5-6}$$

If the average SAD of each pixel is smaller than a threshold, we mark it to be the ground plane. The process is very efficient, particularly for a large textureless region, like a wall. The interest points of patch that lie on the image borders are not taken into account in the match therefore partially visible regions can also be correctly handled.

The global match is only performed on a subset of all color regions, we assume optical axis of the cameras are parallel to the ground plane so only regions are below the horizon line are possible to be ground plane and are performed using this step. After this step, regions (or multiple regions) on the ground plane are marked. Rest regions are performed using the local match.

5.1.3 Plane merging and parameter refinement

After the above steps are applied to the pair of stereo images, the estimations of the 3D structures of all the patches (regions) in the reference image are obtained. If the SAD value is less than a preset threshold, then the patch is marked as *reliable*.

One planar surface may be segmented to several sub-regions. In order to recover meaningful surface structures as large as possible for robots, we try to combine them back to one surface. To solve the problem, first we perform a modified version of the neighboring plane parameter hypothesis approach (Tao et al. 2001) to infer better plane estimates. The main modification is that the parameters of a neighboring region are adopted only if it is marked reliable and the best neighboring plane parameters are accepted only when the match evaluation cost using the parameters is less than a threshold; second, the neighboring regions sharing the same or very close plane parameters are merged into one larger region. This procedure is performed recursively until no more merges occur.

5.2 SMART SAMPLING

Transducing digital video images into displays of a low resolution device is required for the state of art vision prosthesis, including retinal implant and Brainport tongue stimulation device (Brainport 2012). A retinal implant is used to partially restore vision for blind and visually impaired, especially who lost their vision due to retinitis pigmentosa or macular degeneration. By electrically stimulating retinal cells, implants provide low resolution images which may be sufficient to restore light perception and simple object recognition. Currently the state of art retina implants have limited resolution (60 – 100 channels) (Second Sight 2012). Brainport technique invented by Wicab Inc. captures an image and processes the image by converting it into impulses which are sent via electrode array on the tongue (thus tongue simulation) to brain. The brain is assumed to be able to interpret the impulses into visual signals. The tongue simulation has 400 channels (20x20). Both methods are facing a problem of low resolutions, so an important task is how to convey more meaningful information from 2D and 3D images to these devices with their limited spatial resolutions.

In the previous section (Section 5.1), a rapid segmentation-based stereovision approach is used to generate dense 3D maps. It is efficient since it is a feature-based matching approach. The dense 3D map is accurate because it is propagated from accurate 3D measurements of some carefully selected salient features. The outcome of the system is not just a 2D array of individual 3D points (usually produced by a typical stereovision system). Instead, it's a geometric representation of planar surfaces, with geometric relations among neighboring planar surfaces.

With the patch based 3D representation, the performance of visual prosthetic approaches can be improved. McCarthy et al. (McCarth 2011) propose a vision algorithm for retinal implants to support visual navigation. With stereo vision techniques, the system classifies a scene into ground and non-ground surfaces and renders a depth image in a low resolution version (<1000 pixels). The ground area is highlighted so it's easy for perception. The experiments show the down-sampling result in 2-bit and 6-bit dynamic range, but it might miss small/thin objects, such as a pole, a horizontal bar, or a thin tree branch in front of the user, due to use of a uniform sampling.

In this thesis, in order to tackle the limited resolutions of alternative perception, with patched 3D representation, our system provides a number of different alternative perception choices to end users, such as smart sub-sampling, background removal, motion parallax simulation, image re-illuminating and dynamically objects-of-interest highlighting. The end user could select one or more perception methods.

5.2.1 Smart sub-sampling using both color and depth segmentation

A subsampling needs to be conducted to reduce an original 2D/3D map from an original high resolution (Ro) image to a low resolution sampling (Rs) for stimulating visual implants, vibrotaciles or tongue stimulation. Regular uniform subsampling methods sample one pixel every N pixels (N = Ro/Rs). For some thin objects (whose dimension are smaller than N), for example, a lamp pole in front of the blind user, it is impossible to preserve the object after subsampling if the width of the pole in the image is smaller than N. It's dangerous because the user may bump into the objects right in front of him/her.

The smart sub-sampling we have proposed can preserve such thin objects, which are significant by a number of measurements: the distance from the user, the confidence in the 3D measurements, and the shapes. Right now we mostly consider thin but long objects that could be vertical poles and horizontal bars. From the patch based stereo vision method, a 3D map consists of many planar patches with known geometric relations. However, the regular sampling method does not make use of the patch information. The goal of smart sub-sampling is to not lose any important information when the subsampling is performed based on the patch-based 3D representation. Note that we would like to subsample both the color/intensity image (2D map) and its corresponding 3D map.

The smart subsampling is performed on each patch in raster order, while patches with large uncertainty are discarded. Initially the sub-sampled image is filled with blank pixels. During the subsample process, the final sampled 2D/3D map is computed pixel by pixel. A sampled 3D value is filled in a pixel when the pixel is blank; if the pixel is filled in already, then two 3D values are compared and value of the closer 3D is kept. In this way, thin objects in close range can still be preserved during sampling. With the same method, the original color image can be subsampled and any important information can be kept as well.

Fig. 5-2 illustrates the idea using a simulated scene. Fig. 5-2a is a simulated image with a green background (ground plane) and three objects: a building façade, a cubic object, and a thin vertical pole.

Using the patch-based stereo approach, the 3D information of all the regions is obtained (Fig. 5-2b). Fig. 5-2c shows the results of a regular uniform sampling of 20x20 pixels: the thin long pole disappeared. However, using our smart sub-sampling method, the thin pole is preserved in the final 20x20 subsampled image.

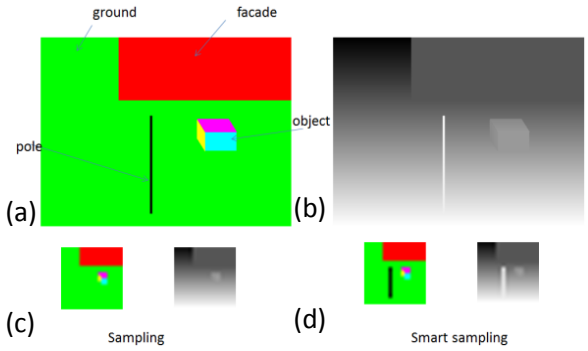


Figure 5-2. (a) The simulated scene with a number of objects (a pole is in a close range); (b) 3D depth map of the simulated scene; (c) sampling results of the 2D image and 3D depth map using a regular sampling method; Note the pole is missing after the regular sampling; (d) sampling results of the 2D image and 3D depth map using our smart sub-sampling method; the pole is still preserved after sampling.

5.2.2 Background removal based on scene labeling

Based on the 3D patches and geometric relations among connected patches in a patch-based 3D representation, simply object detection techniques are applied and the patches can be labeled into different object categories. Background objects, such as static and no-obstacle objects, can simply be removed from the 2D/3D map in order to reduce the complexity of environment scene when presenting to the user, and only the objects of interest, such as persons or obstacles, are “displayed” to the blind user. After removing the background, we only keep small amount of important information in the final sampled 2D/3D map and transduce it into end users, through the special sensor stimulations, such as vibrotactiles or tongue stimulations.

In order to let blind people have stronger ‘feeling’ to observe the obstacles and other types of objects of interest around him/her, the objects of interest (OOIs) can be highlighted by increasing their intensities, whereas the intensities of less-important objects can be decreased so the contrast between OIs and non-

OIs (i.e., background) is increased and the end users are more likely to have a better perception about the OIs. Fig. 5-3a and 5-3b show the results of background removal in both the depth and the color maps; note that only the pole and the cubic box are kept. Fig. 5-3c shows the change of intensity of the box too.

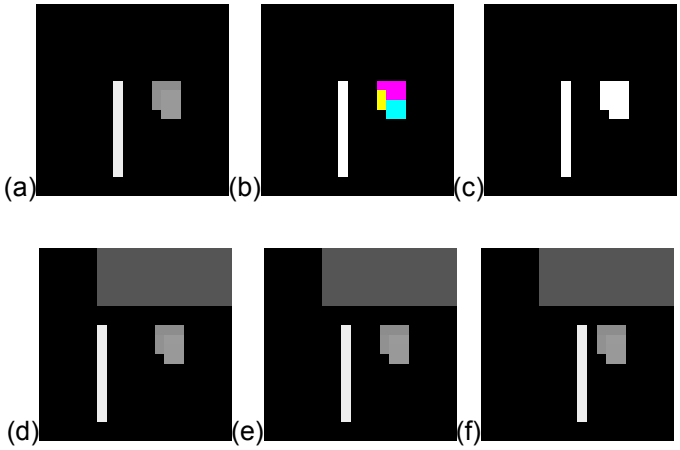


Figure 5-3. (a) The sampled depth image using background removal method; (b) The sampled original image using background removal method (note: the pole and the object are kept in (a) and (b)); (c) The sampled original image using the image re-illumination; (d-f) the object of interest (the pole) in closed range is shifted left and right to simulate its motion parallax.

5.2.3 Motion parallax simulation using 3D

Human is very good at identifying motion. It has been shown that people can recognize objects mostly by their motion from a video sequence, even with low resolution or largely noise-contaminated images in which the objects cannot be easily identified. Therefore, an alternative way to highlight OIs is to generate motion parallax according to the distance of OI from end users, given that we have obtained the range information of these OIs. Since a closer object produces larger motion parallax, by simulating motion parallax of the objects of interest in a small temporal sequence provided to the user, we hope to provide users a strong perception of the close object. In our experiments (5-2. d-f), an OI (the thin pole) is shifted from left to right, which it may produce stronger feeling than just increasing contrasts between OIs and non-OIs. The limitation is that it might be very cluttered if there is more than one object rendered with motion parallax.

5.2.4 Dynamic object re-illumination and highlighting

By constantly changing the intensities of some OIs, we hope to bring more attention from the end users to these objects. The intensities of an OI can be changed from dark to bright and then changed back recursively. Likewise, we can also highlight small but important objects in the close range.

With smart sub-sampling, small objects in closed range can be preserved, but they may only occupy 1-2 pixels in the subsampled image. In order to have a clearer stimulus to an end user, the size of the OI can be changed in the rendering step, that is, it can be enlarged (by a morphological operation) until it reaches a certainty size and then be reduced to the original size; this process can be performed interactively. Alternatively, we can also highlight the contour of an OI dynamically so the alternative stimulation can translate the information better to the user.

5.3 EXPERIMENTAL AND RESULTS

Experiments have been performed to test our approach. Image sequences were captured by the stereovision head Bumblebee, which is fixed on a mobile platform. For a pair of stereo images, the left camera serves as the reference camera. The baseline distance between the left and the right cameras is 12 cm, and focal length of each is 3.8 mm. The stereo system has been pre-calibrated and image pairs rectified.

Fig. 5-4a shows the left view of stereo images captured in an office, and Fig. 5-4b shows the rendered depth map from the estimated planar representations using the patch-based stereo matching algorithm. The plane parameters in the form of “no (a, b, c) d, n”, representing the no. of the plane, the planar normal, the average distance of the plane to the viewer and the uncertainty measurement, are marked for a number of large patches. Note, patches with large uncertainties are highlighted in green.

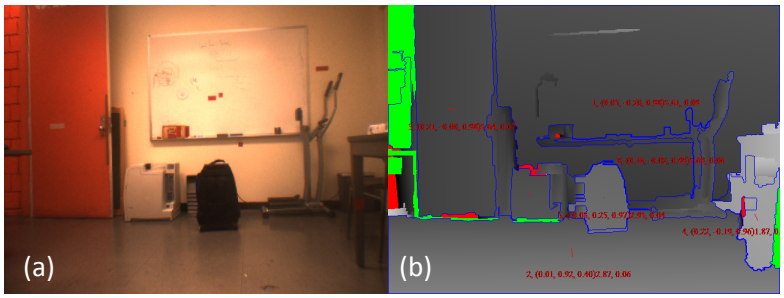


Figure 5-4. (a) Reference image (color image in left view); and (b) 3D depth map generated by patch-based method (the brighter, the closer). For several large regions indexed, the boundaries of regions are marked by closed curves (blue) and planar parameters are drawn on the regions: each arrow and the numbers in a pair of parentheses represent the surface norm, and last number (meters) represents the distance from the surface center to the camera.

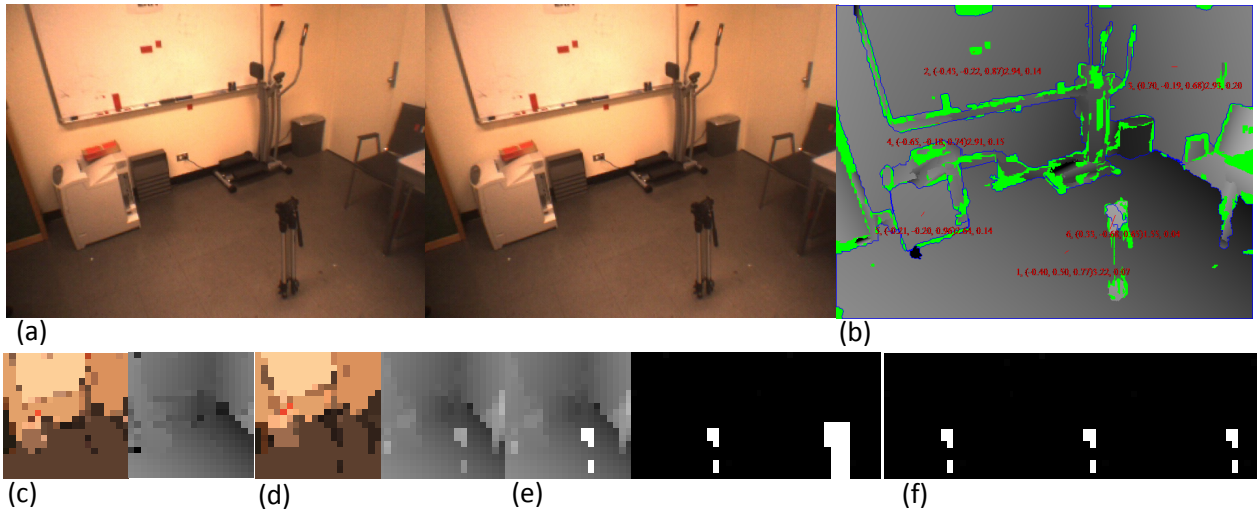


Figure 5-5. (a) A pair of stereo images of an indoor scene captured in an office with a number of objects (note: a tripod is in a close range); (b) 3D depth map of the indoor scene; pixels with large uncertainty are marked in green; (c) sampling results of 2D image and 3D depth map using uniform sampling method: the tripod is missing after regular sampling; (d) sampling results of 2D image and 3D depth map using smart sampling method: the tripod is kept after sampling; (e) sampling results of highlighting objects close to the user, after removing background and applying image dilation; (f) sampling results using motion parallax simulation and only an OI close to the user is shifted towards left, kept original position and shifted towards right, respectively.

In Fig. 5-5a, a pair of stereo images of an indoor scene is shown, including a table, a chair, a printer and a tripod, which are about 1 to 4 meters away from the stereovision head. A depth map (the brighter, the closer) rendered from the results of the plane parametric estimation of the patch-based stereo method is shown in Fig. 5-5b. For several large surfaces, the plane parameters in the forms of “no (a, b, c) d, n” are also shown, with their boundaries highlighted in blue. These plane estimation results are consistent with

the results measured by hand. The parametric representation can be transduced to a blind user easier than an array of depth points (with a uniform sampling method). The geometric representations enable a visually impaired to have safe and efficient navigation.

Fig. 5-5 (c-f) shows results after applying a number of alternative perception approaches. Fig. 5-5c shows that the tripod, which is about 1.55 meters from the user, is missing after uniform sampling, but it is still preserved using the proposed smart sub-sampling approach (Fig 5-5d). In Fig 5-5e, sampling results are shown, after highlighting objects close to the user, removing background and performing dynamic highlighting, consecutively (from left to right). Three samples of the small motion sequence using the motion parallax simulation are shown in Fig. 5-5f; only the object of interest close to the user (i.e., the tripod) is shifted left and right to simulate its motion parallax so that users have better understanding of the 3D distances of OIs.

5.4 SUMMARY AND DISCUSSIONS

We have shown that the patch-based stereo matching algorithm can be applied to more general scenes under the perspective geometry. It is not only an alternative method for stereo computation, but also offers a way to generate content based 3D mosaic. In this chapter, with the perspective geometry, we first use the method to generate a meaningful dense depth map, and then we apply the smart sub-sampling to transduce the important/highlighted information, and/or remove background information, before presenting to visual impaired people. The patch-based method plays an important role in the smart sub-sampling step. It generates surface based 3D depth map instead of 3D point clouds, therefore, it carries more meaningful information and it's easy to convey these information to the visual impaired.

Transducing digital video images into displays of a low resolution device is required for the state of art vision prosthesis. The proposed smart sub-sampling method can preserve close range objects that are significant by a number of measurements: the distance from the user, the confidence in the 3D measurements, and the shapes. A number of practical sampling methods can be used to extract important information and highlight objects of interest in different ways in order to allow an end user to understand the environment easily. One of our ongoing works is to apply the proposed method into

existing visual prosthetic systems, such as retinal implants, vibrotactile, or tongue simulation, to validate the effectiveness of the proposed methods.

Chapter 6 CONCLUSIONS

6.1 SUMMARY

As a summary, the proposed work in this thesis has the following five original contributions.

(1) We extend the previous work on stereo mosaics from static scenes to dynamic scenes, thus allowing the handling of independent moving objects. This is significant in low-altitude aerial video surveillance of urban scenes since traditional methods using change detection fail to work due to motion parallax.

(2) An effective and efficient patch-based stereo matching algorithm has been proposed to extract both 3D and motion information from stereo mosaics of urban scenes, which feature sharp depth boundaries and many textureless regions. This is a unified approach for both 3D reconstruction and moving target extraction. Furthermore, this method can produce higher-level scene representations rather than just depth maps, which leads to our highly compressed content-based 3D mosaic (CB3M) representation. In addition, the new patch-based stereo matching algorithm can also be used with other stereo geometry, such as perspective stereo, for important applications such as visual prosthesis.

(3) We perform thorough experimental analysis of the robustness and accuracy of 3D reconstruction using parallel-perspective stereo mosaics. We show the high accuracy of 3D reconstruction and moving target detection by using a simulated video sequence while both ground truth data of 3D urban model and accurate camera orientation information are available, which motivates us and other researchers for developing robust and efficient algorithms to estimate camera orientation with many image frames. On the other hand, using a simplified camera orientation estimation method for several real-world video sequences, we have found that we can generate very compelling stereo perception and reliable 3D depth information.

(4) A graph-based higher-level scene understanding algorithm is proposed. The method shows the advantage of CB3M data representation in grouping planar patches into higher-level object entities, such as roads, buildings, etc., and in inferring context information. Based on the graph-based 3D planar surface model, other geometric and texture information, further scene understanding can be achieved.

(5) With the patch based stereovision approach under perspective geometry, we propose the smart sub-sampling to transduce the important/highlighted information, and/or remove background information, before presenting to visual impaired people. The patch-based method plays an important role in the smart sub-sampling step since it generates surface based 3D depth map instead of 3D point clouds.

6.2 FUTURE WORK

In this part we will list some possible directions for future research along the work of the thesis. These include: refining 3D modeling, developing and applying the scene understanding algorithm to more real scene data, and fusing local 3D models into a global model.

6.2.1 3D modeling refinement

The proposed system constructs a 3D model from a video sequence using a hybrid approach. From data collection, mosaic generation and up to 3D modeling, each step may include noise and error so it's really challenging to generate a very precise 3D model, especially for some complicated scene, such as the NYC data. Therefore, one of important questions immediately raised is how to refine the 3D model generated from the proposed system.

Because a multi-layer clustering method (in Chapter 4) is used, the understanding to scenes can be constructed hierarchically. Clustering from the surface level to the structure level means we not only obtain geometric information of scenes but also understand scenes into an object level; while clustering from the structure level to the cluster layer means the entire processing of scene understanding goes into a higher level. Therefore, one possible solution is to refine 3D models using the learned scene structure in the object level. Since we constraint urban scenes to only include some primitive object categories, such as building, ground surface, cars and the geometry of objects becomes simpler and many surfaces which does not obtain accurate 3D models might be refined using the above constraints.

6.2.2 Apply scene understanding method on real data

In Chapter 4, we discuss a multi-layer scene labeling method and the experiment on the simulation data is shown. We will need to apply the method on real data, but problem can be more challenging since the

real data is much noisy and the accuracy of the 3D model is lower than that of the simulation data. Therefore the algorithm need to be further developed to deal with real-world issues.

A method proposed to cope with noisy 3D models is to include more object categories in the scene understanding step. Currently only buildings and ground plane are included in the scene understanding step, building and ground recognition may not accurate for all cases. If more dynamic information, such as cars, or traffic flow can be detected, then both accuracy of recognition of building and ground plane could be increased. Chapter 3 shows moving targets (cars) can be detected by our dynamic pushbroom geometry, but some cars may be stationary and can't be detected. So a separate and dedicated procedure for car detection may be included. For example, using the detected vehicles as training samples, a learning-based approach might be applied to detect those static cars. Furthermore, after obtaining more information of traffic flow, the road network/ground plane might be more accurately detection and buildings have better chance to be correctly labeled.

6.2.3 Fusion of local 3D models in perspective view

Chapter 5 shows that the patched based 3D model method is applied on the 3D model in an indoor environment. At one timestamp, a local 3D model is built up. Along the image sequence, multiple local 3D maps are generated using the patch-based method and include lots of redundant information. Therefore, a fast and accurate fusion of multiple local 3D models might be desired.

In order to fuse multiple local 3D maps, accurate camera poses are required. The problem can be solved by structure from motion with sparse feature tracking/matching. SIFT may be used, because unlike other corner based features, SIFT does not select features close to the boundary. Hence using SIFT can ease the occlusion problem usually occurred on the patches' boundaries. SIFT features and the nature primitives (defined along on the boundary of patches) can be complementary to each other. Combining SIFT features with the nature primitives might generate more smooth and reliable fusion result of multiple depth maps.

6.2.4 Testing smart sub-sampling method to existing visual prosthetic

In Chapter 5, we proposed the smart sub-sampling method. One of our ongoing works is to apply the proposed method into existing visual prosthetic systems, such as retinal implants, vibrotactile, or tongue simulation, to validate the effectiveness of the proposed methods. We have worked out a plan to use the development kit of Wicab's BrainPort tongue stimulation, so that we could export the processed results of our techniques to the tongue display unit so that users can test the effectiveness of our approach.

APPENDIX A: PERFORMANCE EVALUATION OF CB3M IN SIMULATION SCENE

We have also compared the final estimated height map with the ground truth data. The error histogram (base 2 logarithmic scaling on the number of pixels) is shown in Fig. A-1a for all the regions (including the moving object regions and other obvious wrong matches). From the error distribution, we have found that the errors of 86.5% points in the reference mosaic are within ± 4 meters. The absolute average value of the errors for those points is only 0.317 meters. Note that in theory, the error of the depth/height estimation by the pushbroom stereo in Eq. 3-2 can be calculated as $\delta Z = (\frac{H}{d_y})\delta y$, where δy is the error in stereo matching (in pixels). In this experiment, H is 300 meters, and dy is from 40 to 320 pixels (from the first pair to the 8th pair of stereo mosaics), and ideally δy is 0.1 pixels with the sub-pixel local match step. Therefore, the theoretical errors after local match go from 0.75 down to about 0.1 meters from the first pair to the 8th pair. However, larger viewing differences introduce larger errors in δy, therefore the error reduction by using larger disparities (from 40 to 320) is not as significant as the theoretical estimation. On the other hand, plane fitting on the multiple interest points with sub-pixel accuracy increases the accuracy in δZ, which leads to a more realistic error range close to the average error of the estimated depths/heights in this experiment (i.e., 0.317m). To show how depth errors vary and how the planar parameters are selected among the eight pairs of stereo mosaics in generating the final height map, Fig. A-1b shows the estimation errors of the planar parameters (from the ground truth) for the 17 largest regions in the reference mosaic. Most of the depth errors are below 0.3 meters, and the magnitudes are comparable among different pairs of stereo mosaics with various “disparities” (i.e., dy). Because of this reason, for each region, we select the “best” result of plane parameter estimation among the eight stereo pairs, instead of using the last pair with the “largest” disparities. Nevertheless, the multi-view approach outperforms the two-view approach significantly. Table 2 compares the average depth errors of depth estimations from a pair of stereo mosaics and all 8 pairs of mosaics for all pixels, 85% of pixels and 75% pixels, respectively. It clearly indicates that multi-view approach reduces the depth errors to about 50%.

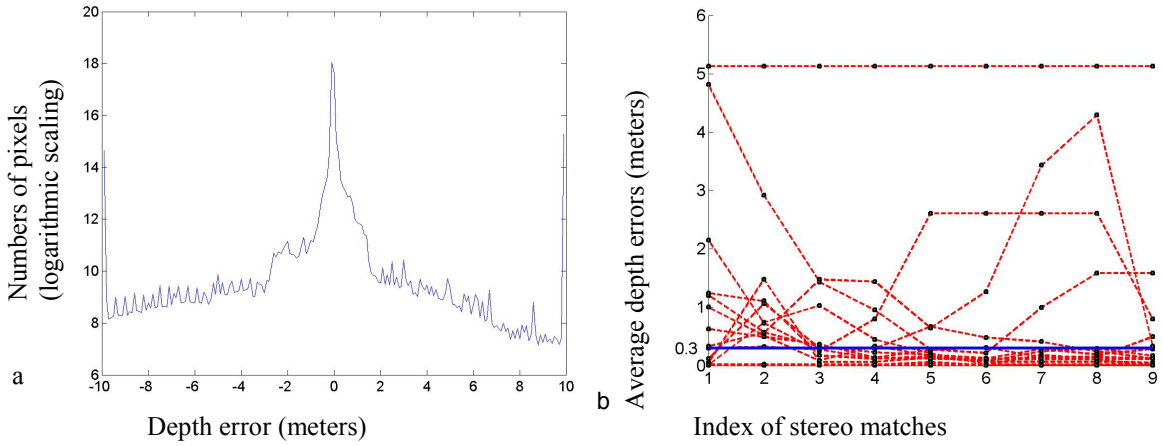


Figure A-1. Depth error analysis. (a) Error histogram. (b) Comparison and selection among the results from the 8 pairs of stereo mosaics for the largest 17 regions. The last column (9th) shows the final selection.

Table 2. Comparison of average depth estimation errors: two views and multiple views

# pixels	75%	85%	100%
$e_{1st\text{-}pair}$	0.20m	0.54m	5.42
$e_{all\text{-}pairs}$	0.07m	0.20m	3.65

After the regions have been merged, we analyze all the reliable regions, and those with obvious 3D anomalies are marked as moving objects traveling along the epipolar lines. For example, in Fig. 3-9a, the heights of the regions labeled 1 and 6, if treated as static objects, are estimated as -39 meters and -50 meters “high” from the ground, respectively, much lower than the ground plane. The small regions labeled 2 and 5 are estimated as 94 meters and 98 meters high from the ground, respectively, much higher than the ground. In fact all these regions only are 2 to 5 meters high from the ground. So these regions with such 3-D “anomalies” if incorrectly treated as static objects are detected as moving targets.

Table 3. Motion estimation errors

Obj Idx	Ground Truth (cm/frame)		Estimated Results (cm/frame)		Errors (cm/frame)	
	Sx	Sy	Sx*	Sy*	ΔSx	ΔSy
1	0	2.485	0	1.649	0	0.836
2	0	-1.499	0	-1.628	0	0.129
3	1.064	-1.262	1.053	-1.08	0.011	-0.181
4	-1.414	1.414	-1.444	1.247	0.031	0.166
5	0	-1.999	0	-2.012	0	0.013
6	0	2.499	0	2.495	0	0.003
7	0.999	0	0.982	-0.076	0.017	0.076
8	-0.781	0	-0.789	-0.178	0.007	0.178

On the other hand, those unreliable regions (as possible candidates for moving objects not along the epipolar lines) further go through 2D-range searches for matches within their neighborhood areas (e.g., 30x30-pixels 2D range). In Fig. 3-9a, regions 3, 4, 7 and 8 are moving targets. They do not obtain reliable matches in the stereo match step, but could find reliable matches from their 2D range searches, between the first mosaic and the rest mosaics. Therefore they are considered as moving targets. Note that those regions marked with red boundaries in the height map have good matches in their 2-D range searches; however, most of them are (1) just at the depth boundaries of dramatic depth changes, and (2) have very small sizes, or have very thin structures, therefore are not considered to be moving targets by using these two criteria. (Same treatment is done for motion detection in real video experiments below.) The estimated motion parameters (sx, sy) (in pixels) of those detected moving targets from the first pair of stereo mosaics are marked on the CB3M map in Fig. 3-9d. The error analysis results of the 8 detected moving targets are shown in Table 2. The average error of the 2D motion estimation is (0.198, 0.008) in velocity (cm/frame), or (0.791, 0.033) in displacements (pixels) between the first pair of the stereo mosaics. The error for the 1st object is the largest since its velocity is not constant.

The compression of a video sequence comes from two steps: stereo mosaicing and then content extraction. For the simulated image sequence, we have 1640 frames of 640*480 color images, so the

data amount is 1.44 GB. The size of pair of the stereo mosaics is 1320*640*2, which has 4.83MB (without compression). The two mosaics in high-quality JPEG format only have 2*75 KB; therefore, a compression ratio of about 9,837 is achieved for the stereo mosaics (the first step). If all the nine mosaics are saved for mosaic-based rendering (Zhu & Hanson, 2006), the data amount will be 9*75KB hence the compression ratio is about 2,186.

Then after color segmentation, 3D planar fitting and motion estimation, we obtained the CB3M representation (Fig. 3-9d) of the video sequence, with the total number of the natural regions N = 1,342 and the total number of boundary points G = 119,477. The total amount of data in its CB3M representation is 80.8 KB (with a header). This real file size is consistent with the estimation of data amount using Eq. (3-24), which is about 79.2 KB (without the coding of the information of neighboring regions for each region; same below). The data amount is reduced to 19.4 KB with a simple lossless Winzip compression on the CB3M data; therefore, the compression ratio is about 76,061:1. Note that the compression ratio depends on how fine the color segmentation is. In the example shown in Fig. 3-9d, the main visual features of the scene are coded. More importantly, the CB3M representation has object contents which can be used for object indexing, retrieval and image-based rendering. The plane parameters (a,b,c,d) for the several representative regions are shown on the CB3M map in Fig. 3-9d (from left to right: one side of a ridged roof, a slanting roof, ground with depth Z= 300.0m, roof of a low building with Z = 289.0m, and side and roof of a tall building with Z=180.0 m), all measured from a camera 300 meters above the ground.

APPENDIX B: COMPUTATION TIME ANALYSIS IN BOTH STEREO MOSAICING AND CONTENT EXTRACTION

The two-phase CB3M construction is also efficient in computation time. The following statistics was obtained when our program was run on a PC with Windows XP, an Intel Core 2 Duo 2.0GHz CPU, 4M cache, 3GB memory, 800MHz FSB (BUS). Most of the computation time in the first phase (stereo mosaicing) was spent on orientation estimation using a pyramid-based image registration method, and stereo mosaicing based on the PRISM algorithm. For a typical video sequence with a resolution of 640*480, the speed of the first phase was about 5 Hz (5 frames per second). More analysis on time complexity of image registration can be found in a related paper of ours (Zhu, et al, 2005).

Since this paper is mainly focused on the second phase, we will provide more information for this phase. In this phase, most of the computation time is spent on two steps: segmentation (a pre-processing step to segment the reference image) and matching (the followed step of matching multi-view pushbroom mosaics). The segmentation step was implemented using the mean shift algorithm by Comanicu & Meer (2002) and a toolbox provided by the authors, and the matching step was implemented by us in C++. Table 4 lists the time performance for the three video sequences we have presented in this paper: the campus scene and the NYC scene. For each sequence, the effective size of each mosaic (denoted as M), the number of patches produced in the reference mosaic after segmentation (denoted as N), the search ranges in both the direction of the camera motion, and the perpendicular direction (denoted as S_h and S_v), the number of pairs of pushbroom mosaics (denoted as K) used in each case, and the times spent in both segmentation and matching are listed in the table. Note that in the table, the sizes of the mosaics are the effective sizes that count the real scene pixels, excluding those pixels that are blank in the borders (this is particularly obvious for the campus scene since the mosaics run in a diagonal direction). Apparently, among the two steps (segmentation and matching), much longer time is spent on multi-view stereo matching, which includes the correlation step in local match (Section 3.3.1), and image warping in match evaluation (i.e., SSD) in the multi-view refinement (Section 3.3.2) and plane updating using global and local constraints (Section 3.3.3). Since both local match and image warping are based on patches over multiple mosaics, the match time is therefore a function of the number of patches N, number of pairs of

mosaics K, and complexity of the scene (leading to various numbers of interests points). Roughly, the time complexity for patch-based multi-view local match and warping can be estimated as

$$T = O(NKS_hS_v) + O(SK) \quad (1)$$

where the first term is for local match, which is proportional to the number of patches, the number of mosaic pairs and the search area, while the second term is for the image warping, which is proportional to the effective size of the mosaic (since all the pixels need to be warped to estimate the goodness of stereo match), and the number of mosaic pairs.

The last two columns of Table 3 are the real time spent in segmentation and matching (in seconds), respectively, and the average time (in ms) spent per patch for segmentation, and per patch per pair of mosaics for stereo match. In particular, the average times in match per patch in the three examples are comparable, which are roughly speaking only functions of the corresponding search ranges. Note that we have not optimized the code for computational efficiency for correlation and warping, which could be implemented using look-up-table and integer iteration techniques that will greatly improve the time performance.

Table 4. Computation time analysis

Clips	Effective Size of mosaic (M)	# of patches (N)	# of mosaic pairs (K)	Search Range (S_h,S_v)	Segmentation time (Ts in seconds, and Ts/N in ms)		Matching time (Tm in seconds, and Tm/(NK) in ms)	
					Ts	Ts/N	Tm	Tm/(NK)
Campus	3900x700	15298	8	(8, 7)	44	2.88	5973	48.81
NYC	3700x2000	37166	3	(30, 8)	330	8.88	9420	84.49

BIBLIOGRAPHY

Boykov, Y., Veksler, O. and Zabih, R. 2001. Fast approximate energy minimization via graph cuts, IEEE Trans. Patten Analysis and Machine Intelligence, Vol. 23, No. 11.

BrainPort Vision Technology, http://vision.wicab.com/technology/, latest update July 2012

Cai, H. Zheng, J. Tanaka, H. Acquiring Shaking Free Route Panorama by Stationary Blurring, IEEE International Conference on Image Processing, 2010, Hong Kong

Chai, J. and Shum, H –Y. 2000. Parallel projections for stereo reconstruction. In Proc. Computer Vision and Pattern Recognition (CVPR'00): II 493-500.

Comanicu, D. and P. Meer, 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Patten Analysis and Machine Intelligence, May 2002

Cornelis,N., Leibe, B., Cornelis, K. and Van Gool, L. 2008. 3D urban scene modeling integrating recognition and reconstruction, Int. J. Computer Vision, 78 (2-3), July: 121-141

Coughlan, C. and Yuille, A. 1999. Manhattan world: compass direction from a single image by Bayesian inference. In Proc. International Conference on Computer Vision (ICCV'99), 941-947.

Deng, Y., Yang, Q., Lin, X. and Tang, X. 2005. A symmetric patch-based correspondence model for occlusion handling. In Proc. International Conference on Computer Vision (ICCV'05), II: 1316-1322.

Dickson, P., Li, J., Zhu, Z., Hanson, A. R., Riseman, E. M., Sabrin, H., Schultz, H.and Whitten, G. 2002. Mosaic generation for under-vehicle inspection. In Proc. IEEE Workshop on Applications of Computer Vision, Dec 3-4

Fusiello, A., Roberto, V. and Trucco, E. 1997. Efficient stereo with multiple windowing. In Proc. Computer Vision and Pattern Recognition (CVPR'97): 858-863

Gupta R and Hartley R, 1997. Linear pushbroom cameras. IEEE Trans. Pattern Recognition and Machine Intelligence, 19(9): 963-975

Hartley R. and Zisserman. A. “Multiple View Geometry in Computer Vision,” 2 ed. Cambridge University Press, 2000.

Herman, T. and Kanade, T. 1984. The 3D MOSAIC scene understanding system: incremental reconstruction of 3D scenes from complex images. Tech. Report, Robotics Institute, Carnegie Mellon University.

Herman, M. and T. Kanade,T.1986. Incremental reconstruction of 3d scenes from multiple complex images. Artificial Intelligence, Vol. 30, pp. 289-341.

Hsu, S. and Anandan, P., 1996. Hierarchical representations for mosaic based video compression, In Proc. Picture Coding Symp., 395-400

Irani, M., Anandan, P., Bergen, J., Kumar, R. and Hsu, S., 1996. Mosaic representations of video sequences and their applications. Signal Processing: Image Communication, vol. 8, no. 4, May.

Kanade, T. and Okutomi, M. 1991. A stereo matching algorithm with an adaptive window: theory and experiment, In Proc. IEEE International Conference on Robotics and Automation (ICRA’91), II: 1088-1095

Ke, Q. and Kanade, T. 2001. A subspace approach to layer extraction, In Proc. Computer Vision and Pattern Recognition (CVPR’01).

Koenen, R.., Pereira, F. and Chiariglione, L. 1997. MPEG-4: Context and objectives. Signal Processing: Image Communications, 9(4):295-300.

Kolmogorov, V. and Zabih, R., 2001. Computing visual correspondence with occlusions using graph cuts, In Proc. International Conference on Computer Vision (ICCV’01), Vol. I:508-515

Koschan, A., Page, D. , Ng, J.-C., Abidi, M., Gorsich, D. and Gerhart, G., 2004. SAFER under vehicle inspection through video mosaic building, Int. J. Industrial Robot, September, 31(5): 435-442

Leung, W. H. and Chen, T. 2000. Compression with mosaic prediction for image-based rendering applications, In Proc. IEEE Int.. Conf. Multimedia & Expo., New York, July.

Li, Y., Shum, H.-Y., Tang, C.-K. and Szeliski, R. 2004. Stereo reconstruction from multiperspective panoramas. IEEE Trans Pattern Analysis and Machine Intelligence, 26(1): pp 45-62.

McCarth, C., Barnes, N. and Lieby, P. Ground surface segmentation for navigation with a low resolution visual prosthesis, in Proc 33rd Annual International IEEE Engineering in Medicine and Biology Society Conference, (IEEE-EMBS), Boston, USA, Aug 2011.

Medioni, G. and Kang, S. 2004. Emerging Topics in Computer Vision. Prentice Hall, ISBN: 0131013661.

Noble, A., Hartley, R. Mundy, J. and Farley, J. 1994. X-Ray Metrology for Quality Assurance, In Proc. IEEE Int. Conf Robotics and Automation (ICRA'94), II pp 1113-1119

Mouragnon, E. Lhuillier, M. Dhome, M. Dekeyser, F. Sayd, P. "3D Reconstruction of Complex Structures With Bundle Adjustment: an Incremental Approach," Proc. IEEE International Conference on Robotics and Automation, pp. 3055-3061, 2006.

Odone, F., Fusiello, A. and Trucco, E. 2000. Robust motion segmentation for content-based video coding, In Proc. 6th Conference on Content-based Multimedia Information Access, College de France: 594-601.

Okutomi M. and Kanade, T., 1993. A multiple-baseline stereo, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 4, pp. 353-363.

Palmer, F., Zhu, Z. and Ro, T., 2012. Wearable Range-Vibrotactile Field: Design and Evaluation, the 13th International Conference on Computers Helping People with Special Needs (ICCHP), July 11-13, 2012, Linz, Austria

Peleg, S., Ben-Ezra, M. and Pritch, Y., 2001. Omnistereo: panoramic stereo imaging, IEEE Trans. Pattern Analysis and Machine Intelligence, 23(3): 279-290

Pollefeys, M., et al. 2008. Detailed real-time urban 3D reconstruction from video, Int. J. Computer Vision, 78 (2-3), July: 143-167

Rav-Acha, A., Engel, G. and Peleg, S. 2008. Minimal aspect distortion (MAD) mosaicing of long scenes. Int. J. Computer Vision, 78 (2-3), July : 187-206

Scharstein, D. and Szeliski, R. 2002. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. Int. J. Computer Vision, 47(1/2/3): 7-42, April-June .

Second Sight, http://2-sight.eu/en/home-en, latest updated July 2012

Shum, H.-Y. and Szeliski, R. 1999. Stereo reconstruction from multiperspective panoramas. In Proc. International Conference on Computer Vision (ICCV'99): 14-21

Smith, R. and Cheeseman, P. "On the Representation and Estimation of Spatial Uncertainty," The International Journal of Robotics Research, vol. 5, no. 4, pp. 56–68, 1986

Strasdat, H. Montiel J. and Davison, A. Real-time Monocular SLAM: Why Filter? ICRA 2010

Sun, C. and Peleg, S. 2004. Fast Panoramic Stereo Matching using Cylindrical Maximum Surfaces, IEEE Trans. System, Man and Cybernetics, Part B, 34, Feb.: 760-765.

Sun, J. Zheng, N. and Shum, H. 2003. Stereo Matching Using Belief Propagation, IEEE Trans. Pattern Analysis and Machine Intelligence, 25(7), July

Tang, H. Zhu, Z. and Xiao, J. Stereovision-Based 3D Planar Surface Estimation for Wall-Climbing Robots, 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, USA

Tao, H., Sawhney, H. S. and Kumar, R. 2001. A global matching framework for stereo computation. In Proc. International Conference on Computer Vision (ICCV'01), I 532-539

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A. 2000. Bundle Adjustment - A Modern Synthesis, In Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science, vol 1883, pp 298-372, eds. B. Triggs, A. Zisserman and R. Szeliski", Springer-Verlag.

Vivet, M. Peleg, S. and Binefa, X. Real-Time Stereo Mosaicing using Feature Tracking, to appear in IEEE int. Workshop in Video Panorama (IWVP 2011), Dana Point, CA, December 2011.

Xiao, J. and Shah, M. 2004. Motion layer extraction in the presence of occlusion using graph cut. In Proc. Computer Vision and Pattern Recognition (CVPR'04)

Zheng, J. Y. and Tsuji, S. 1992. Panoramic Representation for route recognition by a mobile robot. Int. J. Computer Vision, 9(1), pp. 55-76

Zheng, J.Y., and Shi, M. 2008. Scanning depth of route panorama based on stationary blur. Int. J. Computer Vision, 78 (2-3), July :169-186

Zhou, Y. and Tao, H. 2003. A background layer model for object tracking through occlusion. In Proc. International Conference on Computer Vision (ICCV'03): 1079-1085.

Zhu, Z., Hanson, A. R., Schultz H. and Riseman, E. M., 2003. Generation and error characteristics of parallel-perspective stereo mosaics from real video, book chapter in Video Registration, M. Shah and R. Kumar (Eds.), Video Computing Series, Kluwer Academic Publisher, Boston, May: 72-105

Zhu, Z. and Hanson, A. R. 2004. LAMP: 3D layered, adaptive-resolution and multi-perspective panorama - a new scene representation. Computer Vision and Image Understanding, 96(3), Dec: 294-326.

Zhu, Z., Riseman, E. M. And Hanson, A. R. 2004. Generalized Parallel-Perspective Stereo Mosaics from Airborne Videos, IEEE Trans. Pattern Analysis and Machine Intelligence, 26(2), Feb: 226-237

Zhu, Z., Riseman, E. M., Hanson, A. R. .and Schultz, H., 2005. An efficient method for geo-referenced video mosaicing for environmental monitoring. Machine Vision and Applications, 16(4): 203-126

Zhu, Z. and Hanson, A. R. 2006. Mosaic-based 3d scene representation and rendering. Signal Processing: Image Communication, Elsevier, 21(6), Oct: 739-754

Zhu, Z. and Hu, Y.-C., 2007. Stereo Matching and 3D Visualization for Gamma-Ray Cargo Inspection, In Proc. IEEE Workshop on Applications of Computer Vision, Feb 21st-22nd, Austin, Texas, USA