國立臺灣大學電機資訊學院電機工程學系

碩士論文

Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

利用立體攝影機進行色彩與深度感測以達成三維環境重建及物體追蹤

Three-Dimensional Environment Reconstruction and Object Tracking Using RGB-D Sensing of Stereo Camera

洪中易

Chung-Yi Hung

指導教授：連豐力 博士

Advisor: Feng-Li Lian, Ph.D.

中華民國一百零二年七月

July 2013

國立臺灣大學碩士學位論文
口試委員會審定書

利用立體攝影機進行色彩與深度感測以達成三維環境重建及物體追蹤

Three-Dimensional Environment Reconstruction and Object Tracking Using RGB-D Sensing of Stereo Camera

本論文係洪中易君（學號 R00921014）在國立臺灣大學電機工程學系完成之碩士學位論文，於民國一百零二年七月二十六日承下列考試委員審查通過及口試及格，特此證明。

口試委員：

連豐力（指導教授）

簡忠漢

李後燦

黃正民

系主任　顏嗣鈞

誌謝

於臺大兩年的碩士生涯即將結束，這期間中學習及收穫良多，並且不論是在課業、研究及生活上，受到的幫助相當多。最要感謝的是指導老師連豐力博士，在這兩年的耐心教導與提醒，對於研究上需秉持細心及謹慎的態度，並從不同角度及深度看待問題點，進而建立一套有系統的研究方法，以及從聽者的角度設計圖像化簡報。這些訓練不僅在研究上有很大的幫助，相信未來處理任何事情，秉持相同的態度及方法，也能夠迎刃而解；很感謝三位口試委員簡忠漢博士、李後燦博士及黃正民博士耐心地聆聽，並給予相當多寶貴的建議，使本論文能夠更加完善。

感謝 NCSLab 充滿活力且有趣的成員們陪伴，使這兩年研究時光相當充實。感謝志明學長，時常給予研究的想法以及親切的鼓勵，使我在研究及撰寫論文時能夠有所進展；感謝意淳學姊，打理實驗室的大小事務，並給予犀利的想法，讓我能夠從更廣的角度來思考研究或是生活上面臨的問題；感謝多才多藝的俊兆學長，每次聊天都有莫名其妙好笑的梗跑出來；感謝敬凱學長，在剛推甄上臺大時，熱心推薦 NCSLab。詼諧又霸氣的個性帶給實驗室不少樂趣，也讓許多活動能夠順利進行；感謝峻瑜學長，在碩零時帶著我使用立體攝影機，並且給予研究上一些啟發。炒麵是 NCSLab 最難忘的回憶；感謝豐池學長，帶點台味的風格很趣味，是每次歡唱都少不了的 NCS 男高音；感謝峯鳴學長，不論在研究、工作、玩樂及待人處世上都是值得效仿的對象；感謝志祥學長，給了我很多程式上的建議；感謝俊安學長，在討論或是 Meeting 過程中帶來的影像處理分析的知識及建議。也謝謝學長詳細的課程筆記，讓我在上課時能夠更快進入狀況；感謝兩位同學在這段期間一起奮鬥：感謝很有自制力的飛竑，在課業和研究上都是我們三人中的最佳典範。在機器人學上，給予我相當多的知識和建議，使我在陌生的領域下，能夠順利接軌；感謝凱翔，在這幾個月一起度過熬夜奮鬥的夜晚，讓我驚見人類體能的極限。熱血不拘小節的性格，帶給實驗室相當多的歡樂及話題，並時常一起打籃球，使我的球技增進不少；感謝實驗室的新血們，讓碩士生活更加精采。感謝用歌聲征服 NCSLab 的柏逸，教我吉他並擔任實驗室的攝影師；感謝兆良，繼執中學長後讓我再一次看到正拳魂；感謝俊榮教我使用雷射測距儀；感謝詠政教我使用 Kinect，並提供金門特產還有團購餅乾；感謝家維，每次討論都使我獲益良多，並讓我體認到後浪的可怕。

最後，感謝我的家人們，給予我無私的包容以及付出，使我能夠心無旁騖地學習並完成論文；謝謝女朋友慶安，在這兩年的陪伴及關心，並體諒我的忙碌。在此，僅以本論文，獻給所有幫助過我的師長及親朋好友們，謝謝大家。

洪中易 謹誌

中華民國一百零二年八月十五日

利用立體攝影機進行色彩與深度感測以達成三維環境重建及物體追蹤

研 究 生：洪中易　　指導教授：連豊力 博士

國立台灣大學　電機工程學系

中文摘要

三維環境重建是目前一項熱門且應用廣泛的議題，諸如室內環境導覽、虛擬實境以及微創手術之影像導覽系統。立體攝影機同時提供色彩及空間資訊，相較於雷射僅提供空間資訊或單一攝影機提供色彩資訊，更能完整描述環境狀態，提供充足的資訊於三維重建任務上。若能精確地將每一時刻攝影機的相對轉換關係估算出來，立體攝影機量測點便能夠放置在正確的世界座標上，進而建立出三為環境模型。因此首要的任務是利用連續影像上相同特徵點達成立體攝影機的定位。然而，由於立體攝影機的不確定性及錯誤特徵點匹配，不將離群匹配點剃除直接估測攝影機相對姿態將導致定位不精確或是錯誤估測。因此，隨機抽樣一致演算法在此論文中用來作為離群匹配對的剃除。另一方面，由於立體攝影機為被動式感測器，在許多情況如低紋理及光滑材質下，視差影像將產生許多破碎區域，影響三維重建所需的資訊量。因此本論文將提出一個資料前處理的方法，降低量測破碎，進而提高空間重建的品質。

此外，考量到動態環境下建置靜態地圖時，必須將動態物偵測出並將其濾除。因此本論文提出了一套物體偵測及追蹤演算法，以機率形式建立佔據網格地圖擷取出候選物體。接著，候選物體利用 HSV 色彩模型中的色相及飽和度分佈相似性對應到正確的資料庫物體，以解決資料關聯性問題。最後，物體狀態的更新以本論文所提出的更新策略搭配卡爾曼濾波器來達成。實驗結果顯示此系統能夠同時追蹤多重物體，即使物體在一段時間超出攝影機視野或是被遮擋後再被偵測，仍能夠準確追蹤。

關鍵字： 立體攝影機，RGB-D 定位，三維環境重建，物體追蹤，基於可視度之佔據網格。

Three-Dimensional Environment Reconstruction and Object Tracking Using RGB-D Sensing of Stereo Camera

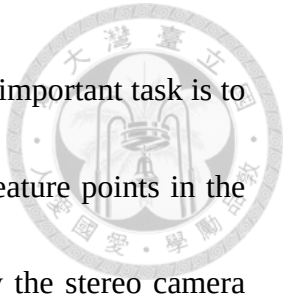
Student: Chung-Yi Hung Advisor: Dr. Feng-Li Lian

Department of Electrical Engineering

National Taiwan University

ABSTRACT

Three-dimensional environment reconstruction is a key technology that has been widely researched over the last decade and has many applications such as indoor environment navigation, virtual reality and visual guidance system for minimal invasive surgery. Stereo camera provides color and spatial information together and therefore is more suitable in 3D environment reconstruction task than other sensors like laser range finder that only provides spatial information or mono camera that only provides color information. Once each camera relative pose is estimated precisely, measurement points provided from stereo camera can be placed at the correct position in the global



coordinate to reconstruct the 3D environment model. Thus, the most important task is to achieve the goal of localizing the camera pose by using the same feature points in the consecutive frames. However, because of the uncertainty caused by the stereo camera noise and the feature point mismatching, estimating the camera pose directly without eliminating the outliers could lead to an inaccurate or wrong result. Therefore, Random Sample Consensus (RANSAC) algorithm is applied to solve the outlier problem in this thesis. On the other hand, because of the limitation of the passive type sensor like stereo camera, the disparity map has many missing data areas that occur in several situations such as measuring object in low textureness or glossy surface. This problem may affect the quality of the reconstructed 3D model. Thus, the data preprocessing method is proposed to enhance the 3D reconstruction quality by reducing the missing data areas.

In addition, considering 3D model reconstruction task in dynamic scene, moving object needs to be detected and removed. Therefore, the object detection and tracking method is proposed to detect an object by constructing the occupancy grid map in probability representation to extract object candidate. Then the distributions of hue and saturation in HSV color space are used to link the candidate to the corresponding database object correctly to solve the data association problem. Finally, the proposed update strategy with Kalman filter is used to renew object states. The experiment results demonstrate that the system can track multiple objects simultaneously and even though

an object is out of the field of view for a while or is in occlusion, the object can still be tracked correctly.

Keywords:

Stereo camera, RGB-D localization, 3D environment reconstruction, object tracking, visibility-based occupancy grid.

Contents

中文摘要 ..... **i**

ABSTRACT ..... **iii**

Contents ..... **vi**

List of Figures ..... **viii**

List of Tables ..... **xii**

Chapter 1 ..... **1**

1.1 Motivation 1

1.2 Problem Formulation 4

1.3 Contribution 6

1.4 Organization of the Thesis 8

Chapter 2 ..... **9**

2.1 Three-Dimensional Environment Reconstruction 9

2.2 Object Detection and Tracking 13

Chapter 3 ..... **18**

3.1 Pin-hole Camera Model 18

3.2 Random Sample Consensus 20

3.3 Image Processing and Description 23

3.3.1 HSV Color Space 23

3.3.2 Morphological Image Processing 23

3.3.3 Connected-Component Labeling 28

3.4 Radial Basis Function 30

Chapter 4 ..... **33**

4.1 Stereo Camera Localization and Mapping 34

4.1.1 Feature Point Extraction 39

4.1.2 Feature Point Matching in Two Consecutive Frames 43

4.1.3 Estimate the relative transformation matrix of rigid body by Least-Squares method using SVD. 46

4.1.4 Camera Pose Estimation with RANSAC Outlier Rejection 51

4.2 Stereo Vision Refinement 58

4.2.1 Forbidden Area Detection and Elimination 59

4.2.2 Holes Detection....66
4.2.3 Dual Orthogonal Linear Interpolation....68
4.2.4 Radial Basis Function....71

Chapter 5....74

5.1 System Architecture....74
5.2 Object Detection....76
5.2.1 Visibility-Based U-Disparity Occupancy Grid....77
5.2.2 Post-processing....85
5.2.3 Object Candidates Bounding Box Extraction....89
5.3 Object Tracking....91
5.3.1 Remove Background Pixels in Bounding Box....91
5.3.2 Registration between Candidates and Objects using Feature Vectors....93
5.3.3 Update Strategy with Kalman Filter....102

Chapter 6....107

6.1 Experimental Hardware....107
6.2 Stereo Camera Localization and Mapping....112
6.2.1 Experimental Scenario Setup....112
6.2.2 The Effect of RANSAC Outlier Rejection Algorithm....116
6.2.3 Relation between Localization Accuracy and Mapping Quality....118
6.2.4 The Accuracy of Feature-based Localization Algorithm....122
6.2.5 Three-Dimensional Reconstruction....125
6.2.6 Mapping Quality and the Proposed Stereo Refinement Algorithm Evaluation....129
6.2.7 Evaluate the Proposed Stereo Refinement in Spatial Aspect....135
6.3 Object Detection and Tracking....148
6.3.1 Object Detection....150
6.3.2 Object Tracking....156

Chapter 7....167

7.1 Conclusion....167
7.2 Future Work....168

References....171

List of Figures

Figure 1.1: RGB-D data structure.....3

Figure 1.2: RGB-D data comparison between stereo camera and Kinect in an outdoor environment. 4

Figure 2.1: Sensor localization categories.....12

Figure 2.2: The object detection and tracking categories. 17

Figure 3.1: Illustration of pin-hole model 19

Figure 3.2: Example of RANSAC algorithm. 21

Figure 3.3: The illustration of the definition of reflection and translation.....24

Figure 3.4: (a) Binary image. (b) Square structure element (SE) with size 3×3 25

Figure 3.5: Process of dilation operator in each step.....26

Figure 3.6: Process of erosion operator. 27

Figure 3.7: Illustration of connected component Labeling for 4-connectivity.....30

Figure 4.1: The proposed system architecture.....34

Figure 4.2: Illustration of the importance of localization for mapping task 37

Figure 4.3: The overall feature-based localization algorithm flowchart.....38

Figure 4.4: An example of spatial feature: Normal Aligned Radial Feature (NARF). 40

Figure 4.5: The flowchart of feature extraction.....41

Figure 4.6: Feature extraction by SIFT detector.....42

Figure 4.7: Block diagram of feature matching processing.....44

Figure 4.8: The result of feature matching by estimate the similarity between two feature descriptor. 45

Figure 4.9: Illustration of two point sets with a certain motion.....49

Figure 4.10: Illustration of the relation between camera pose and relative camera motion. 51

Figure 4.11: Illustration of estimating the relative motion with RANSAC algorithm by two iterations for example. Green circles indicate the feature points in $(k-1)$-th step, while the red dots indicate the feature points in k-th step. Feature points in k-th step are projected by pin-hole camera model with certain transformation matrix.....56

Figure 4.12: The result of using RANSAC outlier rejection algorithm on the matching pairs.....57

Figure 4.13: The block diagram of the proposed stereo refinement algorithm.....58

Figure 4.14: Illustration of the occlusion area that out of the field of view (FOV) of left image plane 60

Figure 4.15: The block diagram of forbidden area detection.....61

Figure 4.16: Illustrate the basic concept of the cutting path extraction according to cutting line. 64

Figure 4.17: The illustration of the proposed forbidden area detection.....65

Figure 4.18: The flowchart of holes detection.....67

Figure 4.19: Illustration of the hole detection concept 67

Figure 4.20: Illustration of hole filling on an arbitrary pixel by DOL interpolation.....69

Figure 4.21: Illustration of the smallest bounding box extraction of a certain hole. 72
Figure 4.22: An example of filling a certain hole using radial basis function interpolation. 73
Figure 5.1: The proposed object detection and tracking system 75
Figure 5.2: Illustration of the visibility-based occupancy grid construction method 79
Figure 5.3: Another example to illustrate the concept of the visibility-based occupancy grid map. 80
Figure 5.4: Each post-processing step applies to the u-disparity occupancy grid. 88
Figure 5.5: Illustration of bounding box extratction from u-disparity obstacle grid. 90
Figure 5.6: Background Pixels Removal for frame #236. 92
Figure 5.7: The registration result of each candidate to the database object. Different objects are enclosed by different color bounding boxes 95
Figure 5.8: Properties of object 1 in frame 236 96
Figure 5.9: Properties of object 1 in frame 191 97
Figure 5.10: HSV histogram of object 2 in frame 236 98
Figure 5.11: Properties of object 1 in frame 237 99
Figure 5.12: Properties of object 2 in frame 237 100
Figure 5.13: A radius distance threshold for the possible range of the object candidates. 101
Figure 5.14: The proposed object update flowchart. 105
Figure 5.15: Two cases of unsuccessful object registration since no measurement in current step. 106
Figure 5.16: Checking if the object is out of the camera field of view (FOV). 106
Figure 6.1: A brief introduction of Bumblebee2 BB2-03S2-60 stereo camera. 109
Figure 6.2: IEEE 1394 Interface 110
Figure 6.3: Hokuyo URG-04LX-UG01 and SICK LMS100 laser range finders. 111
Figure 6.4: Experiment scenario 113
Figure 6.5: Experiment platform and accessories 114
Figure 6.6: Camera path and corresponding image captured from right CCD 114
Figure 6.7: Horizontal laser data. The camera horizontal motion is estimated by applying ICP method to align two consecutive laser data 115
Figure 6.8: Vertical laser data 115
Figure 6.9: Comparing the result of using feature-based localization method with and without RANSAC outlier removal. 117
Figure 6.10: Top view of the camera path. Red cross signs represent the positions estimated by laser-ICP; blue square signs indicate the positions estimated by stereo camera feature-based localization method; green star signs show the given command positions 120
Figure 6.11: Translation and rotation error comparing to laser scanner 120
Figure 6.12: Laser data mapping with localization by stereo feature-based localization method and the given commands. 121
Figure 6.13: Stereo featured-based localization comparing to laser scanners. 123
Figure 6.14: Absolute translation error comparing to laser range finder. 124

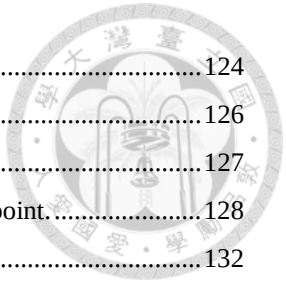


Figure 6.15: Accumulate distance for moving 18 steps. 124
Figure 6.16: The mapping results in each step with two time interval. 126
Figure 6.17: Result of the three-dimensional environment 127
Figure 6.18: The local views of the reconstruction results at the same camera viewpoint. 128
Figure 6.19: The 3D model projects to image plane with certain camera poses 132
Figure 6.20: Illustration of the concept of using PSNR as similarity index to compare the 3D reconstruction quality in the color space viewpoint. 132
Figure 6.21: Comparing the 3D reconstruction result in different case. 133
Figure 6.22: Valid pixels projected from 3D model with and without applying the proposed stereo refinement method. White area in (a) and (b) indicate the valid pixel, while black region represent the pixels without valid value. 133
Figure 6.23: Experiment scenario setup. A plane stands in front of the stereo camera with different view angle 138
Figure 6.24: The target image and the corresponding depth map of Data #3. 138
Figure 6.25: Two orthogonal laser data and transformed to camera coordinate. The local laser data is used to be the input of plane estimation. 139
Figure 6.26: Using local laser data to estimate the plane parameters 140
Figure 6.27: Comparing raw depth map and the depth map generated from the plane. 140
Figure 6.28: The 200×200 rectangular ROI is selected, which is enclosed as in the depth maps. 141
Figure 6.29: The absolute error between the depth of interpolating pixels and the depth generated from laser data. Red lines represent the result of the dual orthogonal linear interpolation approach, while the blue lines indicate the result of radial basis function method 142
Figure 6.30: The data #1 interpolation result of two different approaches. 143
Figure 6.31: The data #2 interpolation result of two different approaches. 144
Figure 6.32: The data #3 interpolation result of two different approaches. 145
Figure 6.33: The data #4 interpolation result of two different approaches. 146
Figure 6.34: The data #5 interpolation result of two different approaches. 147
Figure 6.35: Experimental scenario setup 148
Figure 6.36: Experimental scenario setup 149
Figure 6.37: Detection rate of each object. 152
Figure 6.38: An example of histogram-of-oriented gradients (HOG) method failure detection case. 152
Figure 6.39: For the near range object, one of two cases that is considered to be false detection for example. 153
Figure 6.40: For the near range object, one of two cases that is considered to be false detection for example. 154
Figure 6.41: The far coming object not only has the false detection cases of entering the image plane and the occlusion, it also has the case that is too far to be detected 155
Figure 6.42: Tracking result in image space and Cartesian space 158

Figure 6.43: Object tracking result with applying Kalman filter.....159
Figure 6.44: Object tracking result without applying Kalman filter.....159
Figure 6.45: Too close objects are measured as the same candidate in frame 60. 160
Figure 6.46: The candidate in frame 60 without Kalman filter 161
Figure 6.47: Position of object #1 with and without Kalman Filter 162
Figure 6.48: Position of object #2 with and without Kalman Filter 162
Figure 6.49: Position of object #3 with and without Kalman Filter 162
Figure 6.50: The path is divided into three parts. 164
Figure 6.51: Near path object tracking result comparing to laser. 166
Figure 6.52: Far path object tracking result comparing to laser. 166
Figure 6.53: Circular path object tracking result comparing to laser. 166

List of Tables

Table 4.1: Notations Definition 38
Table 5.1: The definition of some notations for a database object 76
Table 5.2: Bounding conditions of $P(O_U \mid V_U, C_U)$ 82
Table 6.1: The specification of stereo camera BB2-03S2-60 110
Table 6.2: The specification of Hokuyo URG-04LX-UG01 111
Table 6.3: The specification of SICK LMS100 111
Table 6.4: 3D model projected to image plane and compare to target image. 134
Table 6.5: The mean and standard deviation of interpolation errors comparing to laser scanner (m). 142
Table 6.6: Comparing the processing time with different interpolation approaches 142
Table 6.7: Successful detection rate of each object. 152
Table 6.8: The tracking result of near path (m) 164
Table 6.9: The tracking result of far path (m) 165
Table 6.10: The tracking result of circular path (m) 165

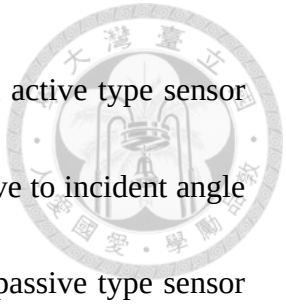
Chapter 1

Introduction

1.1 Motivation

Three-dimensional object and environment model reconstruction is a popular topic that has been researched in the last decade and plays an important role in many area such as robot navigation [1: Henry et al. 2012], virtual reality [2: Marcincin et al. 2012] and visual guidance system for minimal invasive surgery [3: Park et al. 2012].

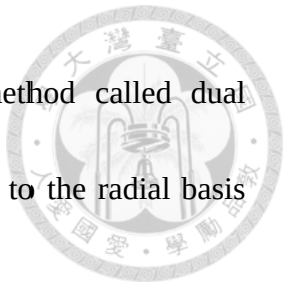
To build a 3-D model, not only the spatial information of an object is needed but also the color information. If the system only has spatial information, the model shape is built without knowing its appearance. Contrarily, if the system has only color information, the model cannot be reconstructed since the points cannot be placed in the correct positions. It shows the importance of color-spatial data structure for 3-D model reconstruction task. Recently, many sensors that have the ability to acquire color-spatial data have been developed, such as stereo camera and Microsoft Kinect. Both of them provide RGB color image and the corresponding disparity map (depth map) with same image coordinate, and this data structure is named as RGB-D data [5: Zeisl et al. 2012]. Moreover, RGB-D data can be extended to a RGBXYZ data structure, as shown in Figure 1.1. Therefore, the whole environment 3D model can be reconstructed by several



frame data capturing in different positions. In addition, Kinect is an active type sensor and its range image is acquired from the IR module, which is sensitive to incident angle and sunlight [6: Suarez et al. 2012]. Contrarily, Stereo camera is a passive type sensor and its range image is estimated by block matching algorithm, and do not have incident angle problem and is less sensitive to sunlight. Figure 1.2 shows the RGB-D data acquired from stereo camera and Kinect in outdoor environment. It is obvious to observe that the number of valid pixels in stereo depth map is larger than the number of valid pixels in Kinect depth map. Hence, stereo camera which belongs to passive type sensor is used in this thesis.

To place these points to the correct positions in the global coordinate, estimating the camera pose in each step is necessary, which is often called localization task. Many existing methods used to localize stereo camera pose have been developed, and one of these approaches is based on tracking the 3-D coordinate of image features and is often called “visual odometry” [16: Scaramuzza et al. 2011].

Although the environment 3D model can be reconstructed by the localization method which tracks the image feature points, many problems still need to be solved. One of the problems affects the mapping quality is the shortcoming of stereo camera itself. The disparity map generated from two CCDs of the stereo camera by local correspondence method has many small missing data areas, which are often called



"broken holes." To fix these missing data, the interpolation method called dual orthogonal linear (DOL) interpolation is proposed and is compared to the radial basis function (RBF) interpolation method in the experiment in Chapter 6.

On the other hand, it is suitable to reconstruct 3D static environment model by the above method. However, if the environment is a dynamic scenario, build the model without removing the dynamic points in the measurement data frames will causes two problems: First, the dynamic points will affect the localization result since they are considered to be the outliers; Second, these dynamic points may occur in the consecutive data frames twice or more, therefore, these points will be mapped into the 3D model several times, causing the ghost effect. In order to detect and avoid the dynamic points, this thesis proposed the object detection and tracking system to track the states of object. Once the states of objects are known, such as the velocity of the object, the points on the object can be seen as dynamic points and be removed.

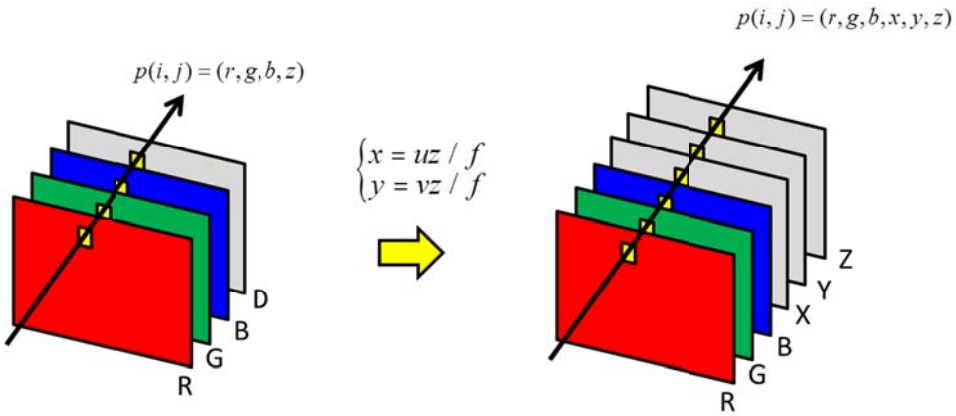
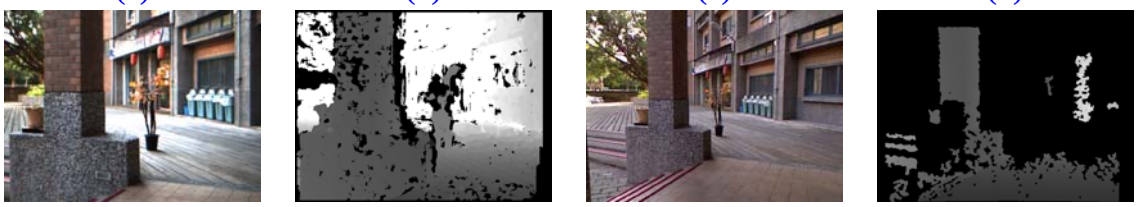


Figure 1.1: RGB-D data structure

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



(e) (f) (g) (h)

Figure 1.2: RGB-D data comparison between stereo camera and Kinect in an outdoor environment.

(a)(b)(e)(f) Color image and depth map from stereo camera.

(c)(d)(g)(h) Color image and depth map from Kinect.

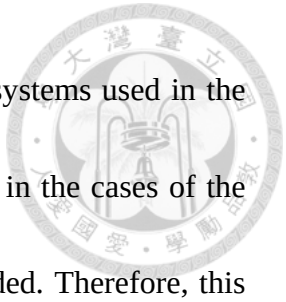
1.2 Problem Formulation

In order to construct 3-D environment model by RGB-D measurements acquired at different positions, the sensor poses in each step need to be known. If the camera poses are known ideally, the data points with RGBXYZ structure can be placed to the correct positions with corresponding colors. However, these sensor poses are usually unknown in practice and need to be estimated. Many researchers investigated the six degrees of freedom sensor poses estimation by aligning two point clouds in k and $k-1$ steps using ICP and ICP variant methods. However, due to the uncertainty of stereo camera, it is not suitable to use ICP and ICP variant directly. Moreover, ICP needs a suitable initial guess or it may return a wrong result due to the fact that ICP aligns two point clouds

into local minimum. Contrarily, since camera moves step by step, many same feature points captured in the consecutive frames. By using the relative 3D coordinates of these points, the camera relative pose can be estimated by least-square method without any initial guess. However, two challenges need to be solved. First, features may not be linked correctly from previous to current frame, considering these wrong matching pairs in motion estimation may cause inaccurate or wrong result. Secondly, due to the sensor uncertainty, point may have inaccurate 3D coordinate, and will also affect the motion estimation result. These two cases are considered to be the outliers, and will be solved by applying Random Sample Consensus (RANSAC) algorithm.

On the other hand, stereo camera is a passive sensor, which the disparity map is generated by finding the same features from reference to target images, which is sensitive to illumination and textureness. This cause many missing data in the disparity map and therefore affects the mapping quality to the 3D models. To overcome the problem, this thesis proposes a data interpolation method to fix the missing data area efficiently by the average of the horizontal and vertical linear interpolation results.

Besides, considering 3D model reconstruction task in a real life scenario, many moving objects that do not belong to the 3D model need to be filtered out. Therefore, the object detection and tracking system is proposed. Many researchers have investigated object detection and tracking based on stereo vision, especially in the field



of intelligent transportation system (ITS). However, these tracking systems used in the traffic scenario do not need to consider tracking an object correctly in the cases of the object returning back to the camera field of view or partially occluded. Therefore, this thesis proposed the object registration method based on color distributions in HSV color space and the update strategy to update the object states to solve the data association problem during the system encounter the cases of object returning back to the camera FOV or partially occluded.

1.3 Contribution

The main contribution of this thesis is that the existing image feature-based localization method, the proposed stereo refinement algorithm, the visibility-grid map construction method proposed in [29: Perrollaz et al. 2012], the proposed object detection method and object tracking algorithm are combined together to achieve two main goals which are three-dimensional environment reconstruction and object tracking using stereo camera.

For the first topic, static environment reconstruction can be divided into two parts, which are sensor localization and stereo data refinement. For the first part, the existing feature-based localization method with RANSAC outlier rejection to achieve the goal of

six degrees of freedom (6-DoF) camera pose estimation [16: Scaramuzza et al. 2011] is integrated in this thesis. After finishing the localization step, each point measured by stereo camera can be added into 3D global model at the correct position with its color, and thus the 3D environment model can then be constructed. However, since many missing data in stereo camera, the proposed stereo refinement method combining with forbidden area elimination, missing data (hole) detection and hole filling is applied to fix this missing data area.

The second topic is the proposed object detection and tracking system. To detect object candidates from stereo camera data, the existing visibility-based occupancy grid map in u-disparity space [29: Perrollaz et al. 2012] with a slightly modification is integrated with the proposed post-processing algorithm in thesis. After object candidates are extracted, the next problem is how to link candidates to the database objects correctly. This problem is so called data association, and is solved by comparing the distributions of hue and saturation channels of the corresponding image patch in HSV color space with the proposed background pixels elimination method. Finally, to update the states of database objects in different cases, this thesis proposed an update strategy to handle the problem with Kalman filter.

The proposed systems can not only be used on stereo camera but also on other sensors which provide the same RGB-D data structure.

1.4 Organization of the Thesis

This thesis has 7 chapters including Chapter 1. The remaining part of this thesis is organized as follows: Literature survey is presented in Chapter 2. The related algorithms are discussed in Chapter 3. Two main parts of this thesis are discussed in the following two chapters. The three-dimensional environment reconstruction methods are shown in Chapter 4, including sensor localization and data refinement algorithms. In Chapter 5, object detection and tracking algorithms are presented. The experimental result and analysis are shown in Chapter 6. In the end of this thesis, the conclusion and future works are presented in Chapter 7 to show the benefits of the main ideas of the proposed system and point out some disadvantages that will be improved and extended.

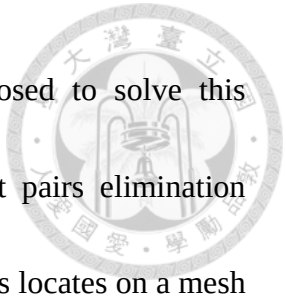
Chapter 2

Background and Literature Review

2.1 Three-Dimensional Environment Reconstruction

In the last decade, many researchers have been investigated on how to reconstruct an environment map precisely using RGB-D sensor. According to the work in [1: Henry et al. 2012], to build a 3D environment map completely, a mapping system should consider three components, which are spatial alignment (localization), close loop detection and global consistency.

For the first component, which is spatial alignment, is the most important element for mapping system to localize the sensor poses. As mentioned in Section 1.3, if sensor does not know its position accurately, the measurements from the sensor cannot map to the correct positions in the 3D global model. Many existing ways to align two consecutive data frames have been developed to achieve the goal of localization method. The traditional and most popular way to align two point clouds is Iterative Closest Point (ICP) method [9: Bsel et al. 1992]. In the ICP registration algorithm, closest point in different point clouds is associated to compute the optimal rigid transformation iteratively that minimizes the mean-square error of each associated point between two datasets. However, due to noise points in the range data that affect the correctness of



point association, many ICP variant related techniques are proposed to solve this problem. For example, [10: Turk et al. 1994] proposed the point pairs elimination mechanism to remove point pairs that are too far apart or either points locates on a mesh boundary to avoid the outliers effect. [11: Chen et al. 1991] proposed point-to-plane error metric instead of point-to-point and get a better result on two surfaces registration. Both of these two variant methods only consider the spatial information. For sensors that generate color point cloud, performing ICP with color constraint can solve the data association problem more convenient. For example, [12: Johnson et al. 1997] proposed the point pairs elimination using hue (the hue channel of HSV color space) of each points as a filter to be a constraint during the closest point search in every ICP iteration. In [13: Men et al. 2011], the method not only consider the hue of each point as an elimination constraint, but includes the hue into the error metric as 4D-ICP, which the 4D means the x, y, z-coordinate and an additional hue intensity. Although many ICP variant algorithms solve the data association problem, both the above ICP and ICP variant algorithms are suffered from initial guess problem since ICP method aligns two data sets to the local minimum. To solve the initial guess, Makadia [14: Makadia et al. 2006] proposed the method to automatically estimate the initial guess and refine the alignment by translating point cloud surface normal vector distribution into orientation histogram, which is called Extended Gaussian Image (EGI). On the other hand, for the

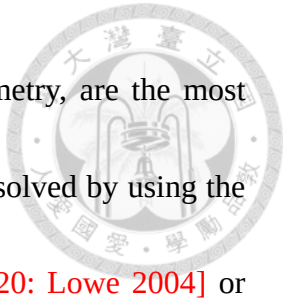


image feature-based localization, which is often called visual odometry, are the most popular to RGB-D type sensors since the initial guess can be easily solved by using the image feature such as Scalar Invariant Feature Transform (SIFT) [20: Lowe 2004] or Speeded-Up Robust Features (SURF) as landmarks [16: Scaramuzza et al. 2011]. However, because many outliers such as wrong feature matching pairs affect the pose estimation result, Random Sample Consensus (RANSAC) outlier rejection algorithm is applied to solve this problem [17: Nister et al. 2004]. Moreover, for binocular stereo vision, since two image planes are fixed, the feature coordinates in reference image plane can be a constraint to check the correctness of each matching pairs of the target image plane in feature matching step. This concept was proposed in [18: Kitt et al. 2010], using the so called trifocal tensor to describe the relationship between three images (which are the two images from previous step and the target image in current step). Besides, [1: Henry et al. 2012] proposed two stage RGB-D localization method by fusing feature tracking with RANSAC outlier rejection and ICP. However, the authors claim that feature-based method is good enough and applying ICP can refine the result slightly. Since image feature-based localization with RANSAC can solve the initial guess and outlier rejection to get a precise localization result and is easily implemented, this thesis chooses this method to achieve to goal of localization.

For the second and third components, which are close loop detection and global

consistency, are used to minimize the error during the frame-by-frame localization. To detect close loop data frames, keyframes are selected and are compared in each data frame [1: Henry et al. 2012]. After detect the close loop, some optimization methods are used to minimize the error. For example, in [1: Henry et al. 2012], two methods are implemented to compare the results: the first method is tree-based network optimizer (TORO) which uses stochastic gradient descent to maximize the likelihood of node parameters subject to the constraints; another is sparse bundle adjustment (SBA), which globally minimize the re-projection error of feature points which are matched in all data frames. Loop detection and global consistency are essential when reconstructing large scale environment model. However, the scenarios in this thesis do not encounter loop closure and global consistency and these problems are considered to be the future works.

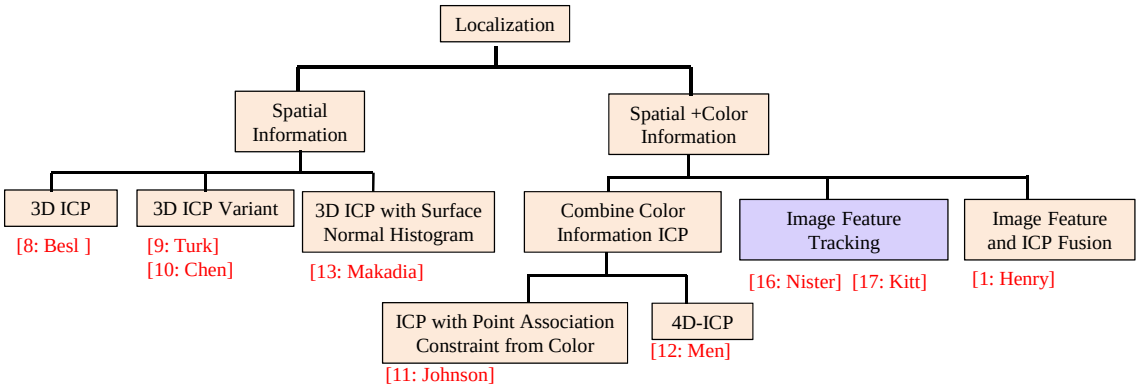
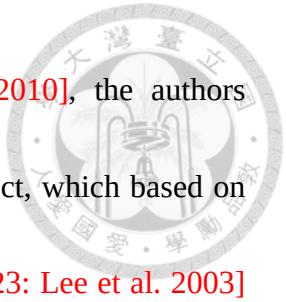


Figure 2.1: Sensor localization categories.

2.2 Object Detection and Tracking

Object detection tracking have been researched for a long time and have been developed by different sensors. According to the properties of different sensors, the object detection task can be categorized into two types, beam-type sensor based and vision-based. For the first category, beam-type sensor, such as laser range finder or ultrasound, provides spatial information by returning an environment point positions. In [34: Wolf et al. 2004], the authors proposed the moving object detection method by constructing a static grid map, and comparing each scan data to this static gird to filter out the dynamic points. However, tracking laser points are a challenge problem, since no other information to determine how to link an object point to the object in next scan correctly. It is well known as data association problem [62: Thrun 2005]. Although many hypothesis approaches have been developed to overcome the problem, considering only spatial information to solve data association problem is still hard and makes ambiguous result.

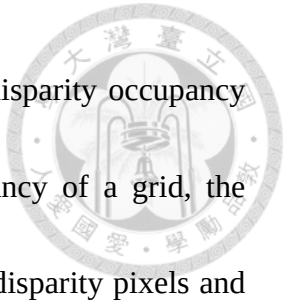
On the other hand, for vision-base category, it can be divided into mono camera and RGB-D type sensor. The main different between these two subcategories is if there has the corresponding range image to the image. Object detection based on mono camera has been researched for a long time since camera provides abundant visual information



to obtain the object appearance. In [22: Saravanakumar et al. 2010], the authors proposed a background subtraction method to retrieve dynamic object, which based on the background modeling performance. To model the background, [23: Lee et al. 2003] proposed using Gaussian Mixture Model (GMM) to model the environment background by several frame images. [24: Barnich et al. 2011] proposed the visual background extractor (ViBe) to achieve better performance than GMM. Both these methods need several images to construct the background, and thus the sensor cannot move too fast. [25: Enzweiler et al. 2009] mentioned that moving object can be extracted by estimating the optical flow of the image to extract moving pixels. The similar concept is tracking features on the object to detect moving object in the image plane [26: Tang et al. 2008]. On the other hand, training-based algorithms are also popular to achieve the goal of detecting specific object. For example, [21: Dalal et al. 2005] proposed using the histograms of oriented gradients (HOG) to detect human based on the edge orientation of the human. [32: Viola et al. 2003] proposes the pedestrian detection method by training the preset pedestrian patterns using Harr wavelet. However, training-based should train a sequence of object patches, and only the specific object can be detected, such as human or vehicle, with different training data.

Stereo camera provides color image with corresponding depth, which has abundant image information and spatial information simultaneously. Therefore, object detection

and tracking can be constructed more easily to combine two different spaces information. To detect object, v-disparity approach is first proposed in [27: Labayrade et al. 2002] and becomes more and more popular. The disparity map is projected to V-disparity space by accumulating the disparity along the v-axis. [7: Hu et al. 2012] and [38: Krotosky et al. 2007] extended the work of Labayrade, the u, v-disparity approach is developed and using Hough transform to extract object bounding box. These methods have a drawback that in some complicate scenario, the line of object bounding box becomes discontinuous in Hough transform line extraction stage. Therefore, some object may not enclose completely by the bounding box. Other approaches based on grid mapping are developed. [31: Oniga et al. 2010] construct a digital elevation map (DEM) to check the height of each grid cell, and construct a density map to check the measurement density of the grid cell. Both of DEM and density map are constructed in Cartesian space. By using these two grid map, the obstacle grid cell can be extracted and find the corresponding object image position by perspective mapping. Although the authors considered the fact that a grid cell at the far distance has less measurement points due to perspective projection by constructing the density grid map, extracting obstacle grid cells by checking the density map is not a complete consideration due to the density of a grid cell may be affected by partially occlusion or missing data. In [29: Perrollaz et al. 2012], Perrollaz et al. proposed the visibility-based occupancy grid map



calculation method for an efficient and formal consideration on u-disparity occupancy grid construction. Instead of using density to describe the occupancy of a grid, the visibility-grid map considers the ratio between the valid number of disparity pixels and the number of disparity pixels that exactly hit (measure) the obstacle to the grid cell and formally uses a probability formula to describe the occupancy of a grid. Based on occupancy grid mapping, tracking an object can be done by Kalman filter [36: Barth et al. 2009] or particle filter [35: Danescu et al. 2012] based on Bayesian framework. However, system encounters data association problem like the situation of beam-type sensor when it tracks multiple object. For example, although the particle tracking method proposed in [35: Danescu et al. 2012] can track multiple objects in most of cases, the tracking result fails when two objects move across each other. In [36: Barth et al. 2009] the authors proposed track-before-detect scheme to solve the data association problem by tracking the image features and then group features by the 3D motion of each feature. In [37: Nedevschi et al. 2007], data association is solved by tracking the features in the object bounding box. These methods can solve data association problem quite well when the object is in the camera field of view. However, these methods may fail when object is viewed from different directions during the object return to the camera FOV. This is because that the feature points are too sparse and too distinctive to describe an object and are not the same in different direction of an object. Contrarily, in

most cases, the hue and saturation distributions of an object in HSV color space do not change dramatically. Therefore, in this thesis, the color distributions of the object are used to be the feature vectors to describe the object without using the feature points.

In this thesis, object detection is solved by slightly modifying the visibility-based occupancy grid construction method proposed in [29: Perrollaz et al. 2012], and data association is solved by using the distribution of the hue and saturation of the object as feature vector. The tracking strategy is proposed to update the state of an object in different situations.

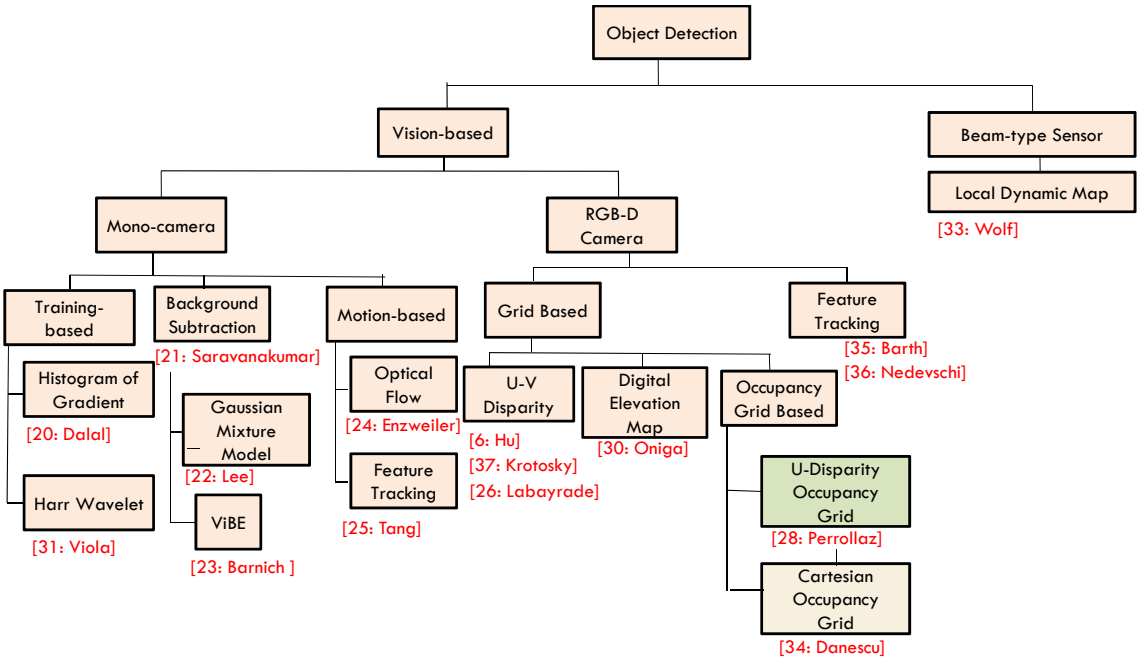


Figure 2.2: The object detection and tracking categories.

Chapter 3

Related Algorithms

3.1 Pin-hole Camera Model

The pin-hole camera model is used to describe the projection of a pinhole camera from 3D coordinate to 2D image plane in mathematics. As shown in Figure 3.1, a 3D point coordinate denoted by (x, y, z) projects to the image plane at the coordinate (u, v), the image center is at the coordinate (u_0, v_0), f is the camera focal length. According to the similar triangles, the pinhole camera projective transform can be written as follows [59: Laganière 2011],

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.1}$$

where s is a scale factor to normalize the projective transform equation. The 3×3 matrix in Equation (3.1) includes all of the camera parameters which are called the intrinsic parameters. f_x is the focal length expressed in horizontal pixels, which is defined as follows:

$$f_x = \frac{f}{px} \tag{3.2}$$

where px is the pixel width. Similarly, f_y is the focal length expressed in vertical pixels, which is defined as follows:

$$f_y = \frac{f}{py} \tag{3.3}$$

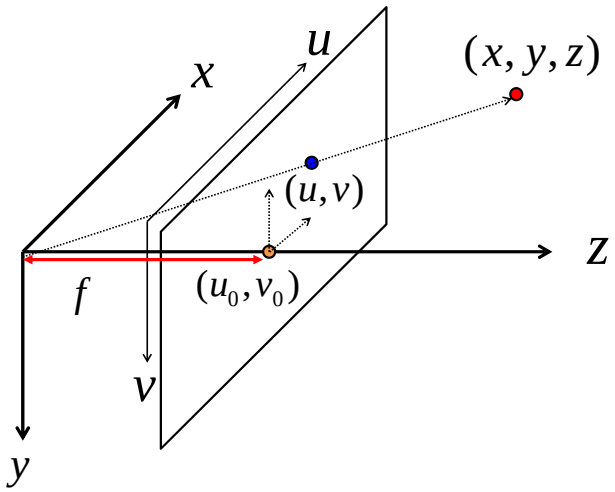


Figure 3.1: Illustration of pin-hole model

Moreover, to generalize the projective transform, the rotation and translation vector are added to the projective transform equation to overcome the problem when the reference frame is not at the projection center of the camera. It can be extended as follows [59: Laganière 2011]:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_X \\ R_{21} & R_{22} & R_{23} & T_Y \\ R_{31} & R_{32} & R_{33} & T_Z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.4}$$

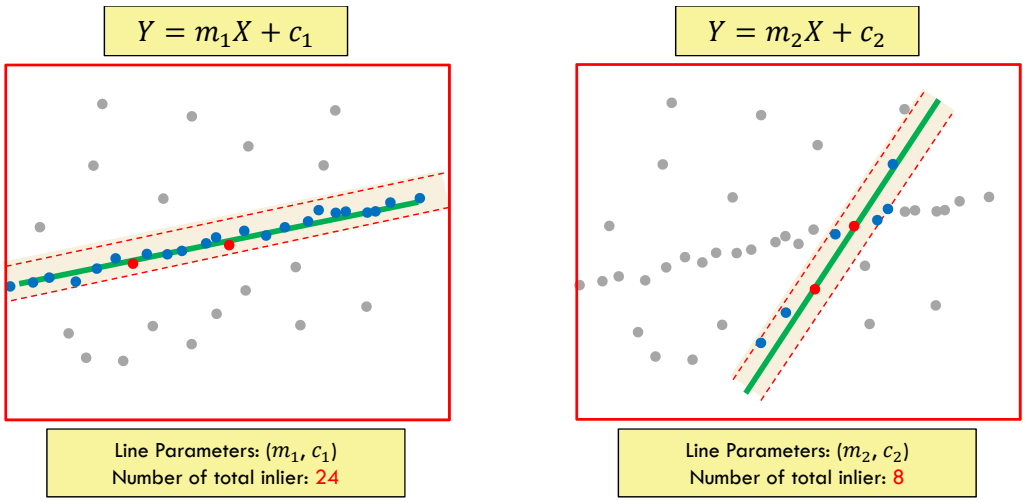
where the elements of the rotation matrix R_{ij} and the elements of the translation vector T_m are put in the same matrix, these elements are called extrinsic parameters of the camera.

3.2 Random Sample Consensus

The random sample consensus is an iterative method to estimate parameters of a mathematical model or transformation from a set of data which contains inliers and outliers and it is well known as its abbreviation, RANSAC [47: Random Sample Consensus from wiki 2013]. Generally, an ideal dataset can be fitted using a certain parameters of the model by least square approach. However, in most cases, data will have noise or wrong measurement due to sensor uncertainty or limitation. Noise or wrong measurements are considered to be the outliers, and the remaining data is called inliers. Therefore, the idea of RANSAC is to find the parameters that are valid for most of the points by discarding the noisy points. The general RANSAC process is listed in Algorithm 3.1.

Figure 3.2 shows an example to illustrate the concept and algorithm of RANSAC method. Assuming that the data set is in two dimensional, that is, each point has coordinate (x, y). The data set is also assumed to have the distribution of a line that can fit the data set. The line model is assumed to be $y = mx + c$. The goal is to find the best parameters (m, c) of the line model that can describe the whole data set. For the first iteration, two points are chosen randomly as red dots shown in Figure 3.2(a). The line parameters (m_1, c_1) can be calculated according to the point-slope formula. The

distances from each point in the dataset to the line can then be calculated. If the distance is smaller than a certain threshold δ, the point is considered to be inlier, shown as blue and red dots in Figure 3.2(a). The remaining gray dots are considered to be the outliers in this iteration. The number of inliers and the corresponding line parameters are stored. In the second iteration, two points are chosen randomly, shown as red dots in Figure 3.2(b). The line parameters (m_2, c_2) are calculated, each of the point-to-line distances are found, and the inliers are counted. The number of inliers and the line parameters in the second iteration step are stored either. In these two iterations, the line parameters in first iteration can describe the dataset with more inliers number than the parameters in the second iteration. For k time iterations, the best line parameters in k-th iteration are chosen if the number of inliers is the largest.

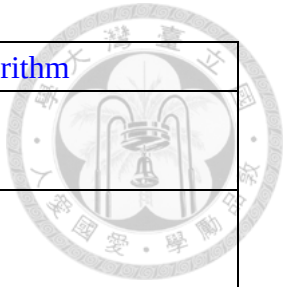


(a) (b)

Figure 3.2: Example of RANSAC algorithm.

(a) The first iteration result with better sample selection.

(b) The second iteration result with worse sample selection.



Algorithm 3.1: General Random Sample Consensus Algorithm

Input: Dataset of points P

Output: Model parameters $Model$

Set the best model $bestModel \leftarrow \phi$
Set the best inlier set $bestInliers \leftarrow \phi$
Set the number of best inlier set $NBestInliers = 0$
Define the number of iterations N
Define model error threshold $threshold$
for $i = 1$ *to* N
  $SampleSet \leftarrow$ Randomly select k points from P
  Compute $CurrentModel$ from $SampleSet$
  $CurrentInliers \leftarrow \phi$
  for all points P_i in P
    Compute the error ε of P_i by using the $CurrentModel$
    if $\varepsilon < threshold$
      $CurrentInliers \leftarrow CurrentInliers + P_i$
    end if
  end for
  Count the number of $CurrentInliers$ $[NInliers_{Current}] = size(CurrentInliers)$
  if $NInliers_{Current} > NBestInliers$
    $bestModel \leftarrow CurrentModel$
    $bestInliers \leftarrow CurrentInliers$
    $NBestInliers = NInliers_{Current}$
  end if
end for

3.3 Image Processing and Description

3.3.1 HSV Color Space

HSV color model separate hue, saturation and value into three independent channels. Each channel of HSV has specific meaning to describe the color. The original color image data are stored in R, G, B three channels, therefore image need to be transformed from RGB color space to HSV color space. The transformation formulas are as follows [45: HSL and HSV from wiki 2013]

$$MAX = \max(R, G, B) \tag{3.5}$$

$$MIN = \min(R, G, B) \tag{3.6}$$

$$V = MAX \tag{3.7}$$

$$S = \begin{cases} 0, & if\ V = 0 \\ 1 - \dfrac{MIN}{MAX}, & other \end{cases} \tag{3.8}$$

$$H = \begin{cases} 0^\circ, & if\ \text{MAX} = MIN \\ 60^\circ \times \dfrac{G-B}{MAX - MIN} + 0^\circ, & if\ \text{MAX} = R \text{ and } G \geq B \\ 60^\circ \times \dfrac{G-B}{MAX - MIN} + 360^\circ, & if\ \text{MAX} = R \text{ and } G < B \\ 60^\circ \times \dfrac{B-R}{MAX - MIN} + 120^\circ, & if\ \text{MAX} = G \\ 60^\circ \times \dfrac{B-R}{MAX - MIN} + 240^\circ, & if\ \text{MAX} = B \end{cases} \tag{3.9}$$

3.3.2 Morphological Image Processing

Morphological image processing is used to refine some sets or reduce some small parts in binary image for example. The language of mathematical morphology is set

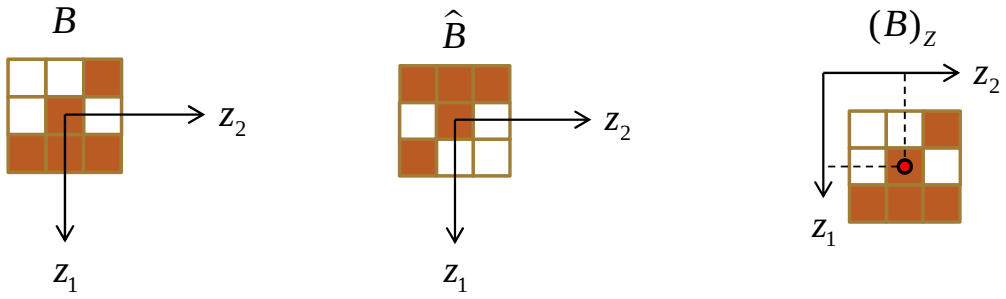
theory [58: Gonzalez & Woods 2008]. In binary image, the sets are members of the 2D integer space Z^2 whose coordinates are the (x, y) of a white pixel in the image. These white pixels are defined as the foreground pixels, whereas the other pixels are called background pixels. Two additional definitions are used extensively in morphology, which are not found in basic set theory, are listed and described as follows. The reflection of set B which is denoted as $\hat{B}$ is defined as follows:

$$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\} \tag{3.10}$$

Figure 3.3(b) illustrates the concept of reflection, which the elements in $\hat{B}$ are equal to the reflecting elements in B. On the other hand, the translation of set B which is denoted as $(B)_z$ is defined as follows:

$$(B)_z = \{c \mid c = b + z, \text{ for } b \in B\} \tag{3.11}$$

Figure 3.3(c) illustrates the concept of translation, which the elements in $(B)_z$ are equal to the elements in B by shifting a coordinate (z_1, z_2).

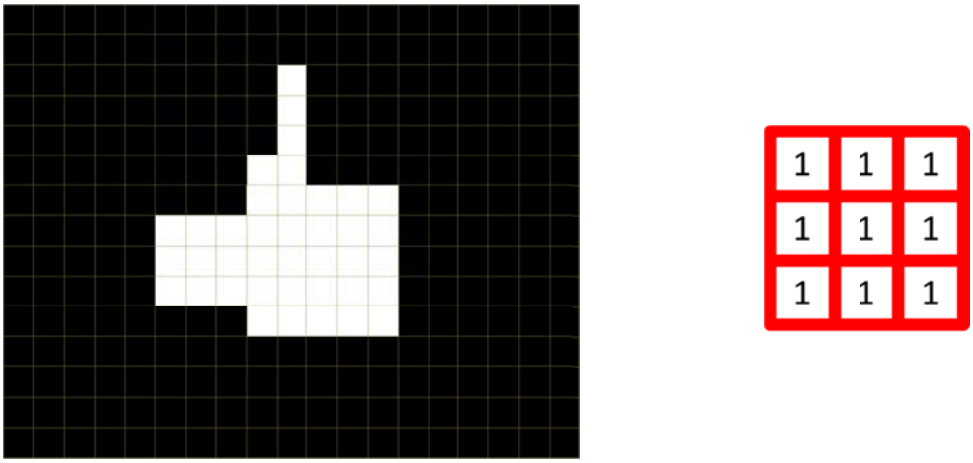


(a) (b) (c)

Figure 3.3: The illustration of the definition of reflection and translation.

(a) The original set B.
(b) The reflection of B.
(c) The translation of B.

Two basic operators called dilation and erosion in the area of mathematical morphology are commonly used to change the shape of the sets in the binary image and are extended to be many advanced operators such as opening and closing. These two operators are described in detail with the binary image and structure element to be an example, which are shown in Figure 3.4.



(a) (b)

Figure 3.4: (a) Binary image. (b) Square structure element (SE) with size 3×3

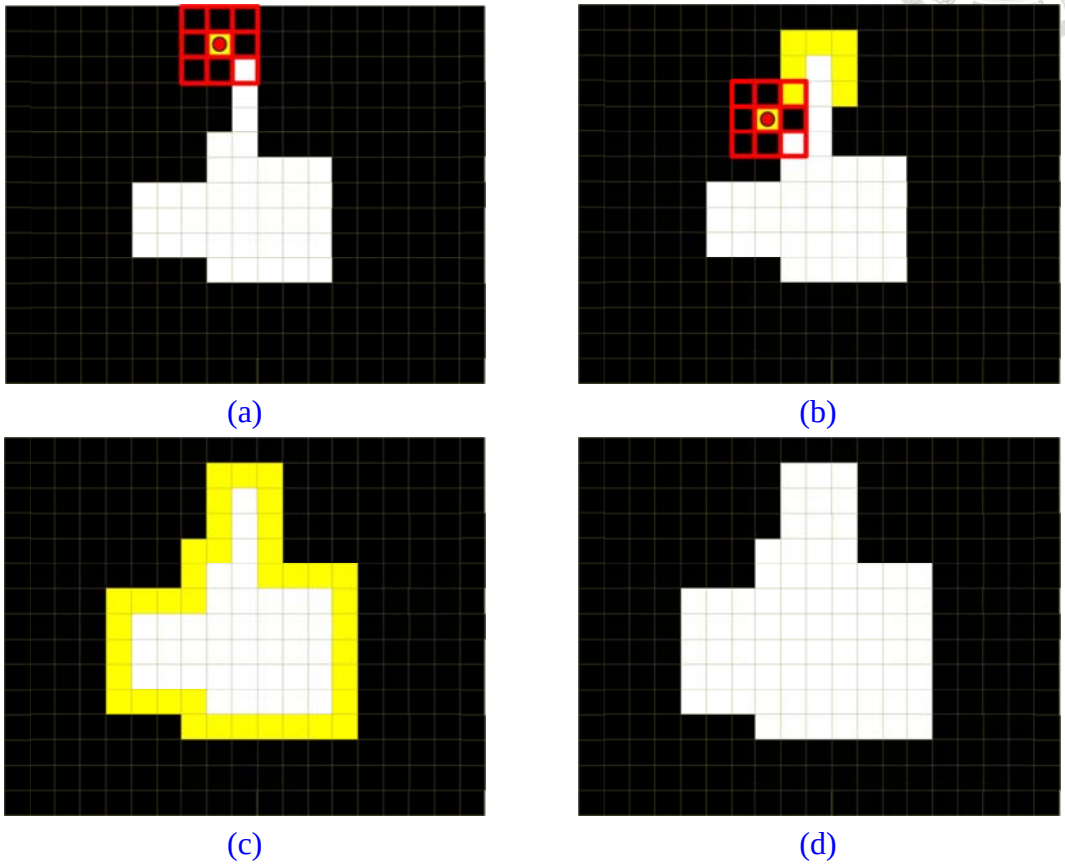
Dilation

The effect of the dilation operator applied on a binary image is to enlarge the boundaries of the foreground pixels which are the white pixels in Figure 3.4(a) by the structure element illustrated in Figure 3.4(b). The definition of dilation operator is as follows:

$$A\oplus B=\{z\,|\,[(\hat{B})_z\cap A]\subseteq A\} \tag{3.12}$$

That is, the pixel (i, j) is marked as foreground if one of pixels in the structure element contacts any foreground pixels in the raw binary image, as shown in Figure

3.5(b). The yellow pixels are the enlarged boundary after applying the dilation operator to the binary image A with structure element B.



(a) (b)

(c) (d)

Figure 3.5: Process of dilation operator in each step.

(a) The center of the structure element scans the image pixel-by-pixel. The position will mark as '1' if one of pixels of the structure element contacts the white pixels of the original binary image.

(b) (c) The yellow pixels are the new added pixels to the binary image.

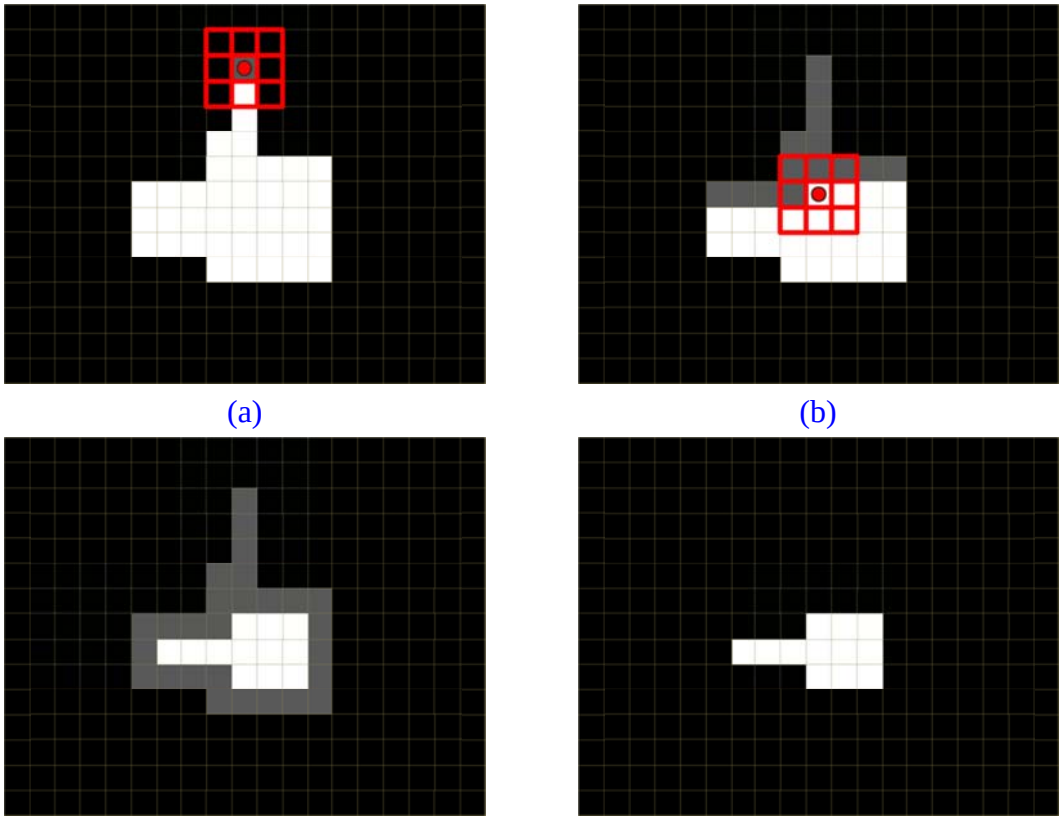
(d) The final result of the dilation process.

Erosion

Contrast to the dilation operator, the effect of the erosion operator applied on a binary image is to erode away the boundaries of the foreground pixels which are the white pixels in Figure 3.4(a) by the structure element illustrated in Figure 3.4(b). The definition of erosion operator is as follows:

$$A \Theta B = \{ z \mid (B)_z \subseteq A \} \tag{3.13}$$

That is, the pixel (i, j) is marked as foreground if all the pixels in the structure element contact the foreground pixels in the raw binary image, as shown in Figure 3.6(b). The gray pixels are the eroded boundary after applying the erosion operator to the binary image A with structure element B.



(a) (b)

(c) (d)

Figure 3.6: Process of erosion operator.

(a) The center of the structure element scans the image pixel-by-pixel. The pixel of the original binary image remains '1' if all the pixels of the structure element contact the pixels of the original binary image.

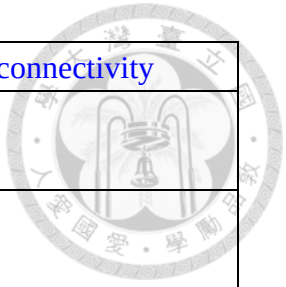
(b) (c) The gray pixels are the subtracted pixels to the original binary image.

(d) The final result of the erosion process.

3.3.3 Connected-Component Labeling

To analyze each morphology region of binary image, it is necessary to distinguish each region at the beginning. Connected-component has the property that each pixel is the neighbor of the other pixels in the region in 4- or 8-connectivity. Connected-component labeling is an algorithm that used to detect connected regions in binary image in computer vision [58: Gonzalez & Woods 2008]. Once the image region is labeled by using connected-component labeling algorithm, many region properties such as area (pixel number), smallest bounding box vertexes and component pixels list can be extracted.

Many ways to achieve connected-component labeling task have been developed. Here a simple algorithm in recursive version is described in Algorithm 3.2 and illustrated in Figure 3.7. In this thesis, the connected-component labeling and region properties can be found in MATLAB using the instruction *regionprops*.



Algorithm 3.2: Simple Connected-Component Labeling with 4-connectivity

Input: Binary Image *Image*

Output: Connected-Component Labeling Array *ConnectedImage*

```
1:   [ImageRow, ImageCol] = size(Image);
2:   ConnectedImage = zeros(ImageRow, ImageCol);
3:   NumberLabel = 0
4:   for i=1:ImageRow
5:     for j=1:ImageCol
6:       if Image(i, j) == 1 and ConnectedImage(i, j) == 0
7:         NumberLabel = NumberLabel + 1;
8:         ConnectedImage(i, j) = 1;
9:         ConnectedImage = CheckNeighbor(i − 1, j, Image, ConnectedImage);
10:        ConnectedImage = CheckNeighbor(i + 1, j, Image, ConnectedImage);
11:        ConnectedImage = CheckNeighbor(i, j − 1, Image, ConnectedImage);
12:        ConnectedImage = CheckNeighbor(i, j + 1, Image, ConnectedImage);
13:      end if
14:    end for
15:  end for

16:  function CheckNeighbor(iIdx, jIdx, ConnectedImage)
17:    if Image(iIdx − 1, jIdx) == 1 and ConnectedImage(iIdx − 1, jIdx) == 0
18:      ConnectedImage = CheckNeighbor(i − 1, j, Image, ConnectedImage);
19:    end if
20:    if Image(iIdx + 1, jIdx) == 1 and ConnectedImage(iIdx + 1, jIdx) == 0
21:      ConnectedImage = CheckNeighbor(i + 1, j, Image, ConnectedImage);
22:    end if
23:    if Image(iIdx, jIdx − 1) == 1 and ConnectedImage(iIdx, jIdx − 1) == 0
24:      ConnectedImage = CheckNeighbor(i, j − 1, Image, ConnectedImage);
25:    end if
26:    if Image(iIdx, jIdx + 1) == 1 and ConnectedImage(iIdx, jIdx + 1) == 0
27:      ConnectedImage = CheckNeighbor(i, j + 1, Image, ConnectedImage);
28:    end if
29:    return ConnectedImage
30:  end function
```

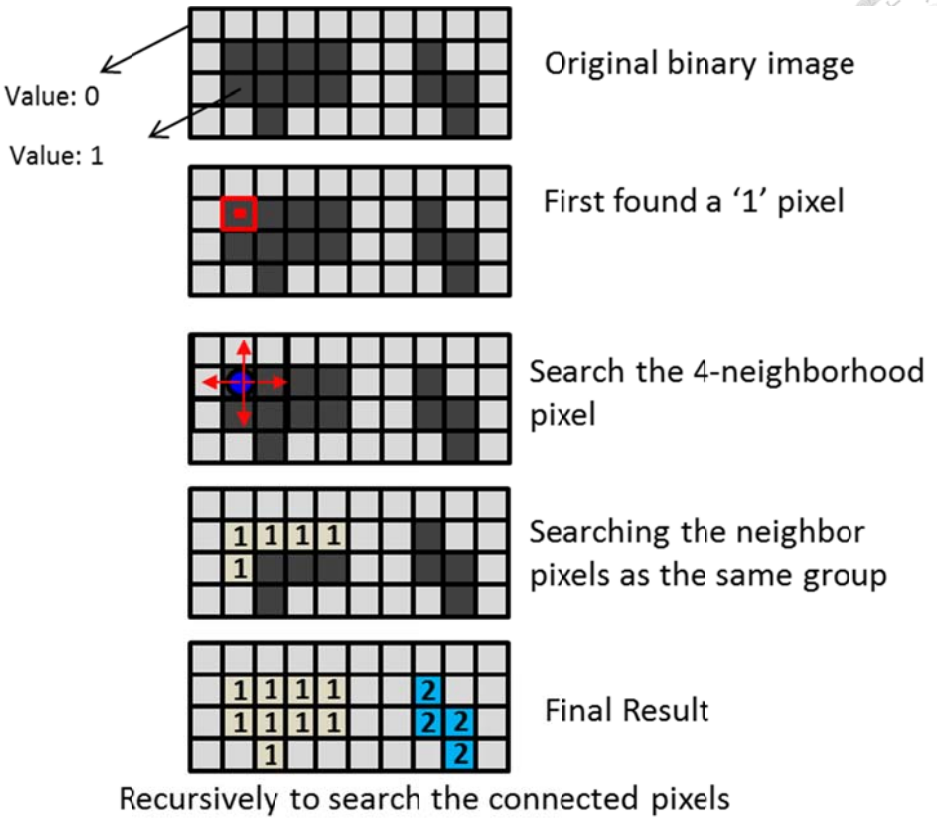


Figure 3.7: Illustration of connected component Labeling for 4-connectivity.

3.4 Radial Basis Function

Radial basis function (RBF) is an interpolation method for calculating unknown data within the range of the available known data points. According to [63: Buhmann 2003], radial basis functions are typically used to construct a function approximately by the form:

$$y(x) = \sum_{i=1}^{N} \omega_i \cdot \varphi(\|x - x_i\|) \tag{3.14}$$

where $y(x)$ is the approximating function of sum of N radial basis functions, each associated with different center x_i and weighted with corresponding coefficient ω_i.

$\varphi(\|x - x_i\|)$ is the so called radial basis function, which has many types, for example:

1. Gaussain:

$$\varphi(r) = e^{-(\varepsilon r)^2} \tag{3.15}$$

2. Multiquadric:

$$\varphi(r) = \sqrt{1+(\varepsilon r)^2} \tag{3.16}$$

3. Inverse multiquadric:

$$\varphi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}} \tag{3.17}$$

where $r = \|x - x_i\|$ is the distance between x and x_i.

To obtain the goal of RBF interpolation, it is necessary to determine a proper radial basis function $\varphi(r)$. After determining the radial basis function, the next step is to train the weights of RBF ω_i. Since each known data point has corresponding known output, the training process takes all the known data point x_j to associate each data x_i to calculate the corresponding weight ω_i, that is,

$$y(x_i) = b_i,\ i = 1...N. \tag{3.18}$$

$$y(x_j) = \sum_{i=1}^{N} \omega_i \cdot \varphi(\|x_j - x_i\|) = b_j,\ j = 1...N \tag{3.19}$$

where x_i is the known node and b_i is the corresponding known output. Extending the Equation (3.19), it becomes:

$$\begin{bmatrix} \varphi(\|x_1 - x_1\|)\omega_1 + \varphi(\|x_1 - x_2\|)\omega_2 + \ldots + \varphi(\|x_1 - x_N\|)\omega_N \\ \varphi(\|x_2 - x_1\|)\omega_1 + \varphi(\|x_2 - x_2\|)\omega_2 + \ldots + \varphi(\|x_2 - x_N\|)\omega_N \\ \vdots \\ \varphi(\|x_N - x_1\|)\omega_1 + \varphi(\|x_N - x_2\|)\omega_2 + \ldots + \varphi(\|x_N - x_N\|)\omega_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \quad (3.20)$$

Since $\varphi(\|x_j - x_i\|) = \varphi(\|x_i - x_j\|)$, the Equation (3.20) can be rewritten by replacing the radial basis function $\varphi(\|x_j - x_i\|)$ to $A_{i,j}$, that is,

$$\begin{bmatrix} A_{1,1}\omega_1 + A_{1,1}\omega_2 + \ldots + A_{1,1}\omega_N \\ A_{2,1}\omega_1 + A_{2,1}\omega_2 + \ldots + A_{2,1}\omega_N \\ \vdots \\ A_{N,1}\omega_1 + A_{N,1}\omega_2 + \ldots + A_{N,1}\omega_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \quad (3.21)$$

By rewriting the Equation (3.21) in matrix form, Equation (3.21) becomes:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & \cdots & \cdots & A_{1,N} \\ A_{2,1} & A_{2,2} & \cdots & \cdots & \cdots & A_{2,N} \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & A_{j,i} & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ A_{N,1} & A_{N,2} & \cdots & \cdots & \cdots & A_{N,N} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \quad (3.22)$$

For simplicity, Equation (3.22) can be expressed as Equation (3.23)

$$A\omega = b \quad (3.23)$$

If the matrix A is nonsingular, the coefficient ω can be obtained by Equation (3.24).

$$\omega = A^{-1}b \quad (3.24)$$

Therefore, due to ω_i is known and $\varphi(r)$ is defined previously, the approximating function can be obtained.

Chapter 4

3D Environment Reconstruction

In this chapter, the proposed 3D environment reconstruction method using stereo camera is presented. The proposed system can be divided into two parts, which are localization and stereo refinement. The overall system architecture can be illustrated by Figure 4.1. For camera at k-th time step, two independent processes perform in parallel. To estimate camera relative movement between current and previous step, feature points captured in these two steps need to be the inputs of localization part to achieve the goal of feature matching process. Some wrong matching pairs cause inaccurate localization result are eliminated by the outlier rejection method based on Random Sample Consensus (RANSAC). For stereo data refinement, wrong measurement pixels in forbidden area are removed by statistic method. The missing data areas which are often called a hole in the remaining disparity map are detected by using connected-component labeling technique. Finally, the missing data regions are filled by dual orthogonal linear interpolation method.

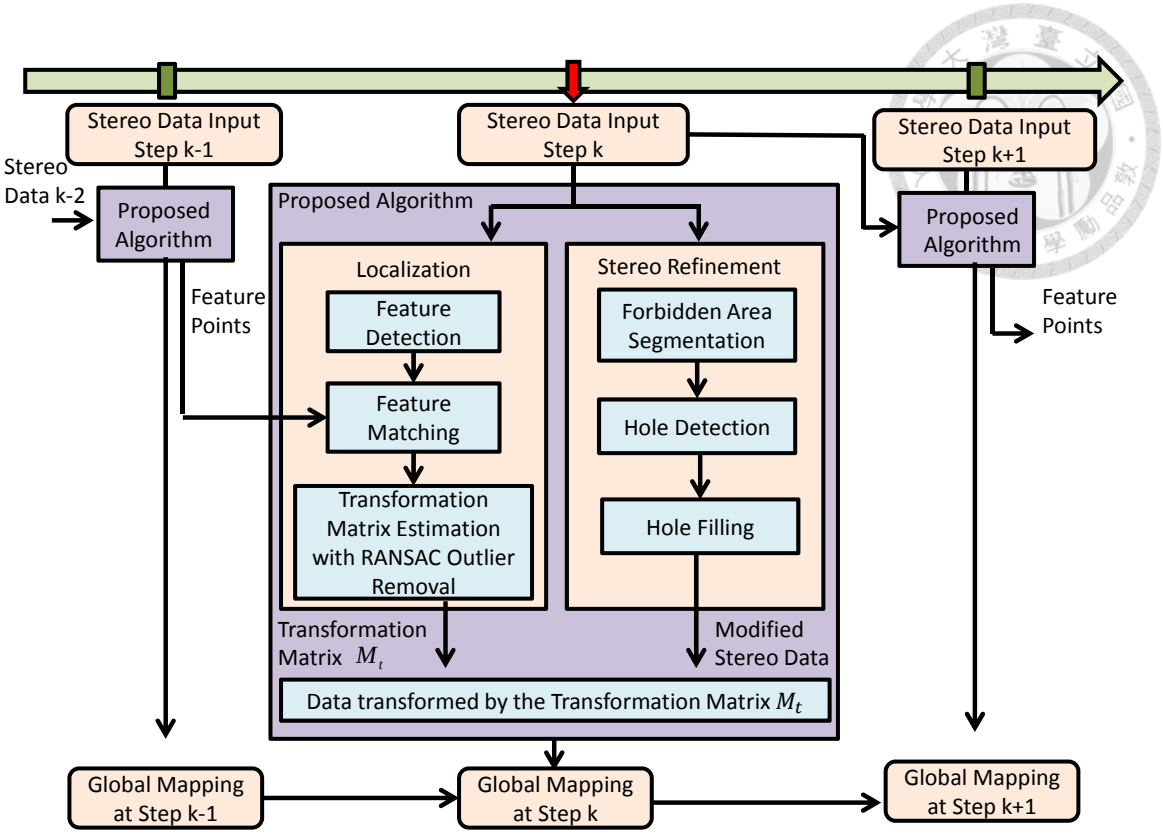


Figure 4.1: The proposed system architecture.

4.1 Stereo Camera Localization and Mapping

One of the advantages of stereo camera is the data structure that combines spatial and color information to a pixel in the image coordinate, that is, $P_i = (x_i, y_i, z_i, r_i, g_i, b_i)$. This data structure provides essential information to 3-D reconstruction. Assuming a camera captures a sequence of the local data in the environment with known camera poses, the global environment 3D model can be reconstructed by placing these data in correct positions to world coordinate. Figure 4.2 shows the concept: two landmarks in the environment are captured by two consecutive camera frames with relative camera

pose T as in Figure 4.2(a). The landmarks are seen by the camera in k-th and $(k-1)$-th steps as in Figure 4.2(b) and Figure 4.2(c) respectively. If the transformation T is known, two measurements in k-th and $(k-1)$-th steps can be aligned together, and therefore the environment can be reconstructed, as illustrated in Figure 4.2(d).

However, in practice the measurement data is acquired from the camera coordinate without knowing the camera pose in world coordinate. That is, the measurement data points cannot be placed to the correct positions in world coordinate. Fortunately, two consecutive images capture the same landmarks in the environment, as illustrated in Figure 4.2(e) and (f) for example. Assuming the environment is a rigid body, a certain transformation which includes rotation and translation can align these landmarks in current camera coordinate to the landmarks in previous camera coordinate correctly. In other words, the transformation can be found out by using the relationship between the corresponding landmarks in current and previous camera coordinates as illustrated in Figure 4.2(g) and (f). By doing so, the 3-D points in current step can be mapped to the previous coordinate. This transformation is the relative camera pose that is necessary to be estimated, and the transformation estimation process is familiar with the term “localization”. In ideal, the camera relative motion can be estimated by simply applying the above concept. However, some error matching pairs affect the motion estimation as shown in Figure 4.2(j) and (k). If the motion is estimated inaccurately, the environment

cannot be reconstructed precisely as illustrated in Figure 4.2(l). Therefore, motion estimation with outlier rejection by random sample consensus (RANSAC) is used in this thesis.

The relation between time flow and the each processes of the localization algorithm is shown in Figure 4.3. For k-th step, stereo camera provides two images from right and left CCDs and then calculates the depth map with respect to the target image coordinate at first. The second step is to detect image feature from the target image. In the third step, image features in k-th step are matched to the corresponding features in $(k-1)$-th step.

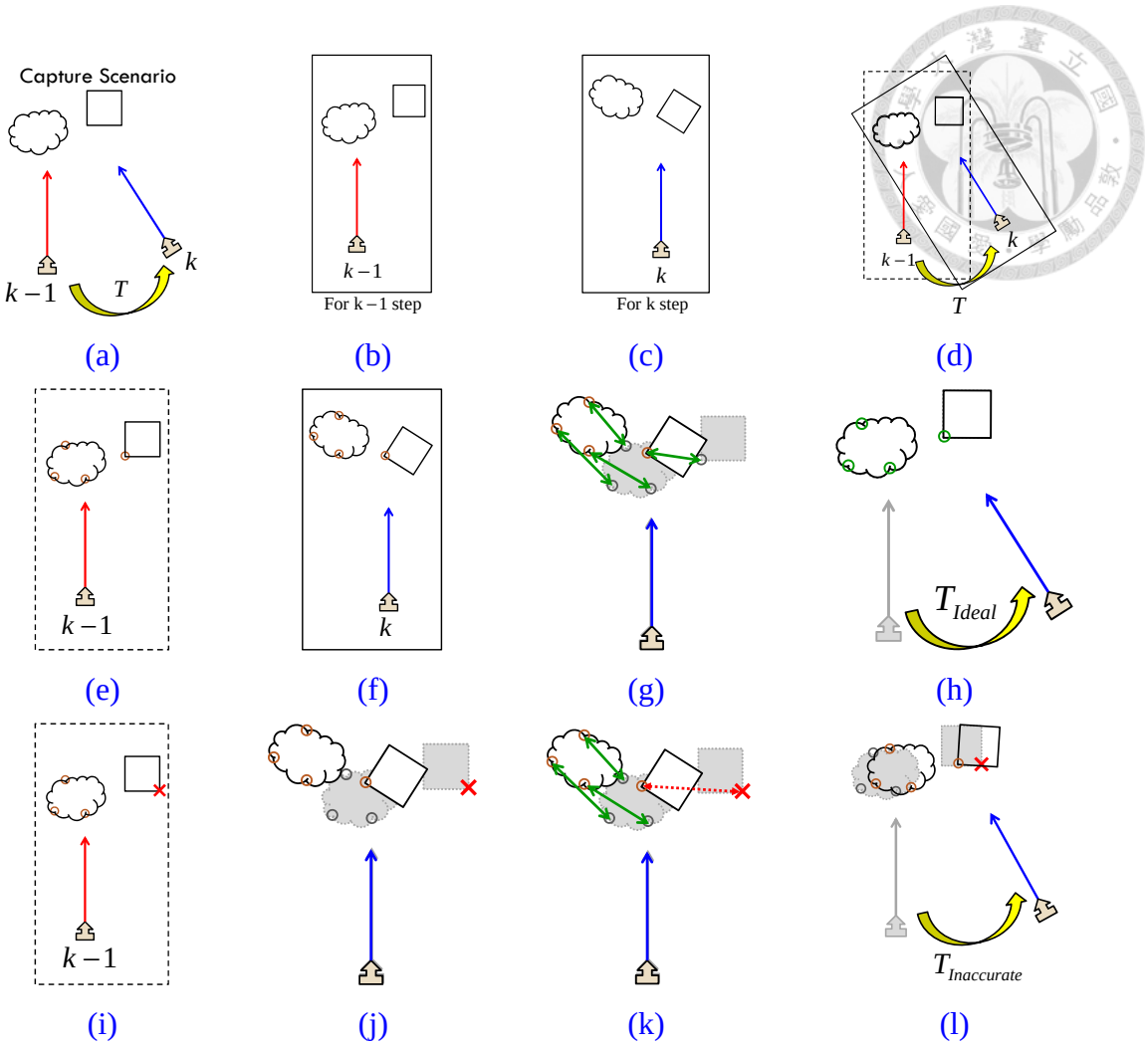


Figure 4.2: Illustration of the importance of localization for mapping task

(a) The environment is captured by camera in consecutive step.

(b)-(c) The camera measurement in $(k-1)$-th and k-th steps in camera coordinates.

(d) If the transformation is known, measurements in k-th step can be transformed to the $(k-1)$-th step camera coordinate to obtain the environment reconstruction process.

(e)-(f) Feature points extracted in $(k-1)$-th and k-th step in ideal.

(g) Overlap these measurements in the same camera coordinates. Features in (e) and (f) are matched by estimating the similarity of the local appearance of each feature.

(h) Since the same landmarks captured by the camera in consecutive time steps are matched well, the transformation relation can be estimated by using these landmarks relative positions.

(i) Feature points extracted in $(k-1)$-th step with wrong feature point extraction for example.

(j)-(k) In practice, there might have some wrong matching pairs that will cause inaccurate or incorrect motion estimation.

(l) If the camera relative motion is estimated inaccurately, the environment cannot reconstruct precisely.

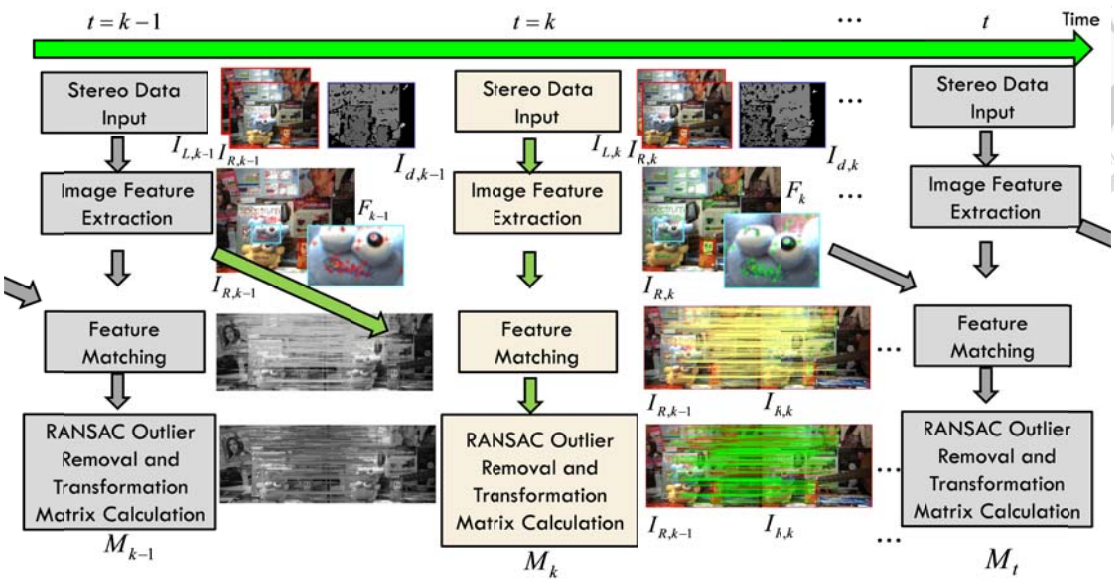


Figure 4.3: The overall feature-based localization algorithm flowchart.

For better explanation, some notations are defined and listed in Table 4.1.

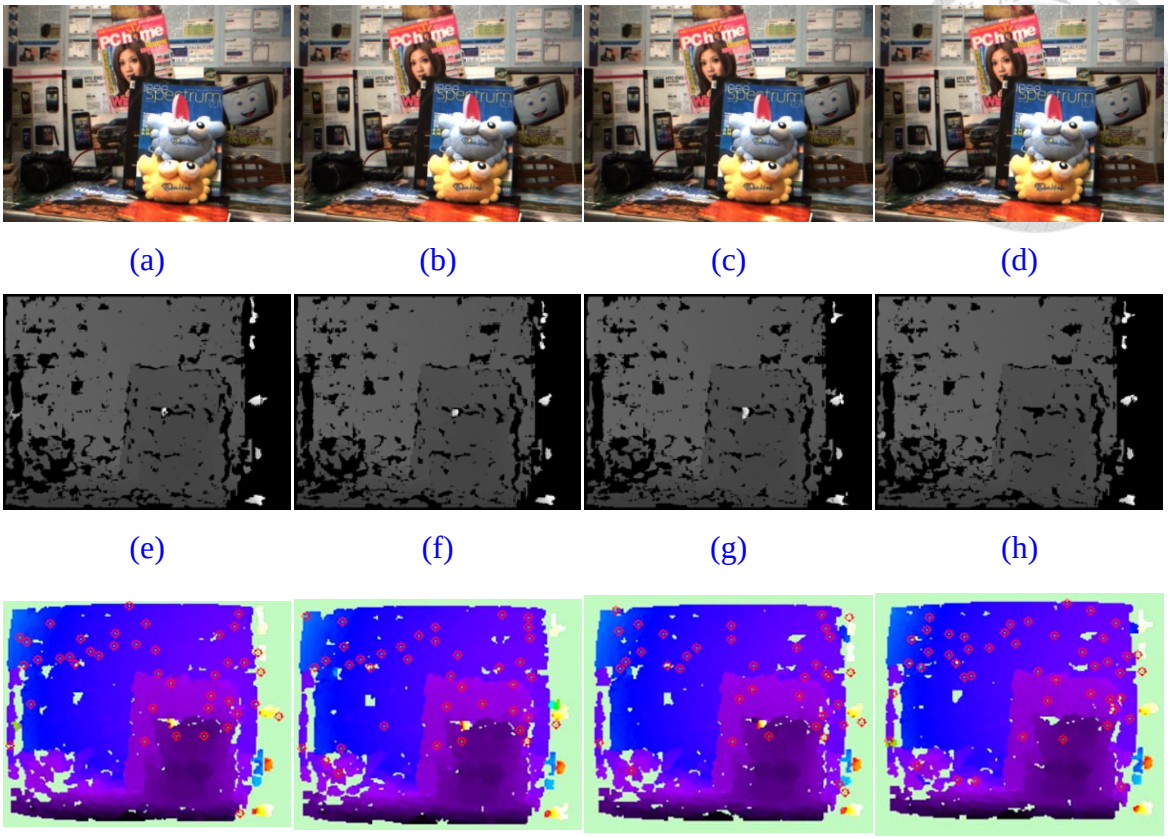
Table 4.1: Notations Definition

Image	
$I_{R,k}$	The right image captured at k-th step
$I_{L,k}$	The left image captured at k-th step
$I_{Tar,k}$	The target image which is defined by different stereo camera system. In this thesis, the target image is equal to right image, $I_{Tar,k} = I_{R,k}$.
I_d	The disparity map corresponding to the target image I_{Tar}
$I_{d,k}$	The disparity map corresponding to the target image $I_{Tar,k}$ at k-th step
Feature	
F_k	The feature set of features extracted from $I_{Ref,k}$
$F_{k,i}$	The i-th feature in F_k
$FP_{k,k-1}$	The feature pair set matched between F_k and F_{k-1}
$FP_{k,k-1,i}$	The i-th feature pair in feature pair set $FP_{k,k-1}$
Point	
$PFP_{i,k}$	The spatial position of a feature in set k of i-th matching pair $FP_{k,k-1,i}$, that is, $PFP_{i,k} = X_{i,k} = (x_{i,k}, y_{i,k}, z_{i,k})$
$PFP_{i,k-1}$	The spatial position of a feature in set k of i-th matching pair $FP_{k,k-1,i}$, that is, $PFP_{i,k-1} = X_{i,k-1} = (x_{i,k-1}, y_{i,k-1}, z_{i,k-1})$
Camera Pose	
C_k	The set of camera pose at k-th step
$M_{k,k-1}$	The relative camera motion from $(k-1)$-th to k-th step.

4.1.1 Feature Point Extraction

The purpose of the feature point extraction step is to detect some parts of the scene observed by stereo camera in the environment for feature tracking. Currently, many types of features that can be used to describe the environment have been developed. In order to identify a point located at different image plane position frame by frame, a feature should be robust or so called “salient” in the field of computer vision to distinguish each data point. Point feature can be categorized into two main types, that is, color and spatial spaces.

Stereo camera provides two images from different image planes and then the disparity map is estimated by using these images, which is secondhand information and using spatial type feature such as Normal Aligned Radial Feature (NARF) [42: Steder et al. 2011] might be unreliable for stereo vision. Figure 4.4 shows the NARF detection results from the depth maps at the same position but in different time steps for example [55: NARF feature from PCL 2013]. Although the frame data are acquired by stereo camera in the same poses, the NARF features are not robust due to stereo uncertainty and missing data problem, as shown in Figure 4.4(i)-(l). Therefore, the Scalar Invariant Feature Transform (SIFT) which belongs to color type feature point is used in this thesis.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4.4: An example of spatial feature: Normal Aligned Radial Feature (NARF).

(a)-(d) Color image captured at the same position in different time steps.

(e)-(h) Corresponding depth maps in different time steps.

(i)-(l) NARF features extracted from range images in different time steps. Red circles indicate the locations of NARF features.

In feature extraction step, a set of salient points F_k is detected at k-th frame in the target image $I_{Tar,k}$. Each feature point has two important data, which are the position in image coordinate $Position_{Image}(F_{k,i}) = (v_{k,i}, u_{k,i})$ and the 1×128 descriptor $Descriptor(F_{k,i}) = D_{k,i} = [d_{k,i,1}, d_{k,i,2}, \ldots, d_{k,i,128}]$ extracted from its local patch. The descriptor is used to determine the correspondence of certain feature in two different images. Moreover, with original SIFT feature information and the spatial information provided from stereo camera depth map, the overall data structure of a feature point provides the

image position of the feature point $Position(F_{k,i})$, descriptor $Descriptor(F_{k,i})$ and the spatial position $Position_{spatial}(F_{k,i}) = [x_{k,i}, y_{k,i}, z_{k,i}]$ in camera coordinate. However, due to the limitation of stereo camera, not all the feature points have depth information because of missing data by occlusion or lighting problem in the left image as shown in Figure 4.6(d). Since invalid disparity is set to 256 (corresponding depth is set to 0), using these features to estimate camera pose may cause wrong result. Therefore, feature points without depth information must be removed before doing camera pose estimation step, as illustrated in Figure 4.6(f). The block diagram of the total feature extraction process is shown in Figure 4.5.

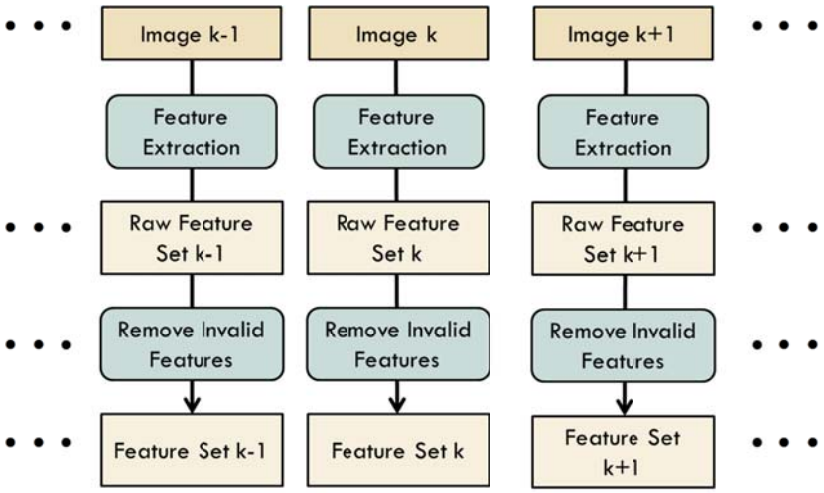
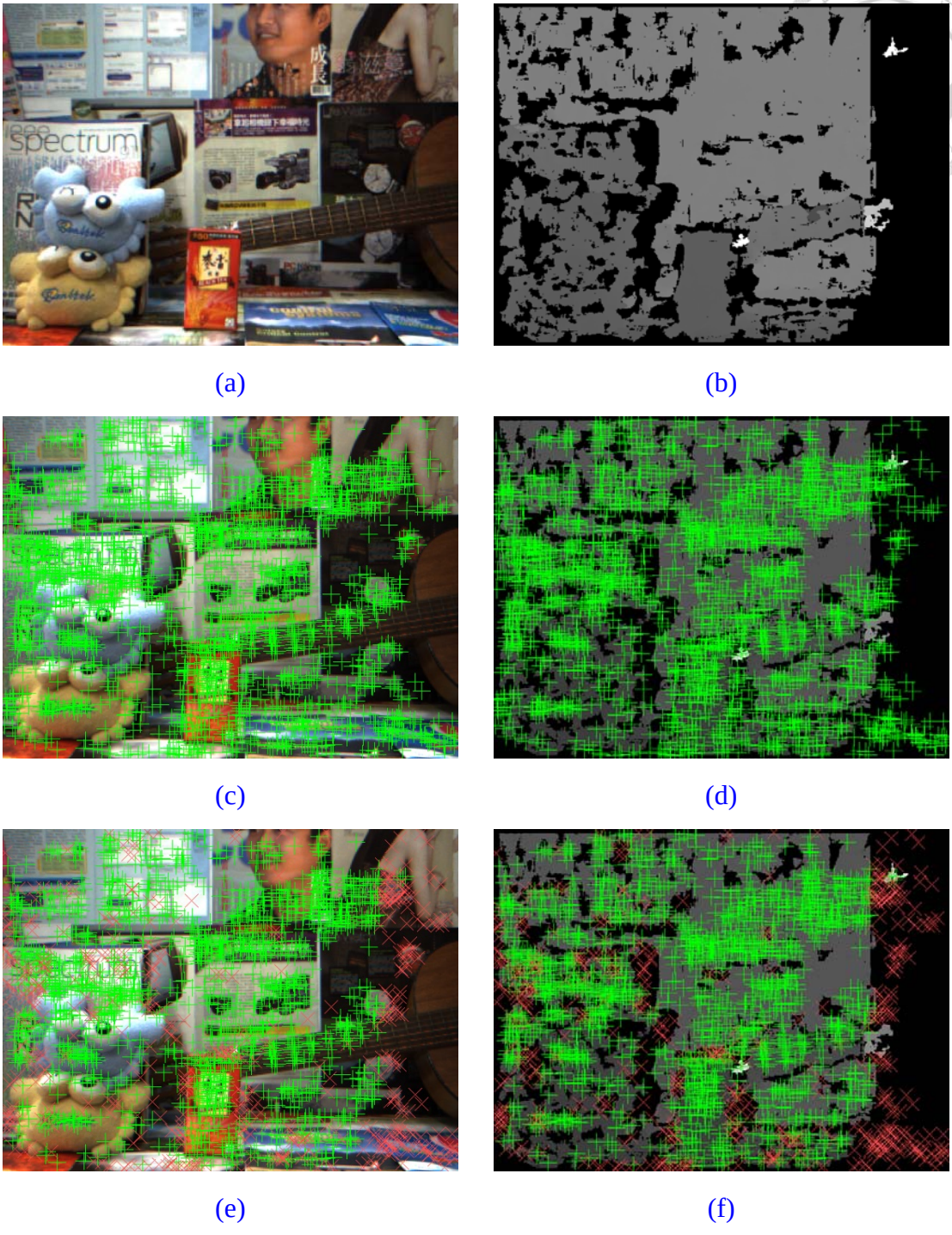


Figure 4.5: The flowchart of feature extraction



(a) (b)

(c) (d)

(e) (f)

Figure 4.6: Feature extraction by SIFT detector

(a)-(b) Input image and Corresponding depth map

(c)-(d) SIFT features with input image and corresponding depth map.

(e)-(f) Features with non-depth information are marked as red cross sign according to the depth map.

4.1.2 Feature Point Matching in Two Consecutive Frames

Several salient feature points are extracted from each target image frame in the stage of feature detection presented in Section 4.1.1. In order to track 3D positions of the features captured in consecutive images, this section aims at recognizing each feature located at different images and formed as feature point pairs. This process is often called feature matching, and the data flow is illustrated in Figure 4.7. For the consecutive images at k-th and $(k-1)$-th step, feature sets F_k and F_{k-1} are extracted as the inputs of feature matching stage, and the output is the feature matching pairs formed as $2 \times N$ matrix $FP_{k,k-1} = \left[FP_{k,k-1,1}, FP_{k,k-1,2}, \ldots, FP_{k,k-1,j}, \ \ldots \ , FP_{k,k-1,N} \right]$, where $FP_{k,k-1,j}$ is the j-th matching pair with 2×1 dimension that stored the indexes of two features $F_{k,m}$ and $F_{k-1,n}$.

A traditional way to achieve the goal of feature matching for certain feature in current step is to compare the similarity of feature points from previous feature set. Since the descriptor represents the local appearance of a feature, the similarity can be evaluated by comparing the point descriptor in current frame to the descriptors in previous frame. Many ways are used to calculate the similarity of two data, such as Euclidean distance used in this thesis. The Euclidean distance measures the square root of sum of square difference of each element between two n-dimension vectors, which is defined as follows.

$$d(p,q)=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2} \tag{4.1}$$

Substituting the descriptors $D_{k,i}$ and $D_{k-1,j}$ into Equation (4.1), the equation becomes:

$$d(D_{k,i},D_{k-1,j})=\sqrt{\sum_{m=1}^{128}\left(d_{k,i,m}-d_{k-1,j,m}\right)^2} \tag{4.2}$$

where $D_{k,i}$ is the descriptor of i-th feature in $F_{k,i}$, and $D_{k-1,j}$ is the descriptor of j-th feature in $F_{k-1,j}$. Therefore, the feature with the smallest Euclidean distance in descriptors is considered to be the successful matching feature pair of $F_{k,i}$ and $F_{k-1,j}$ in F_{k-1}. That is, for i-th feature in $F_{k,i}$, the corresponding feature in F_{k-1} is,

$$\arg\min_{j=1:J}\left(d(D_{k,i},D_{k-1,j})\right)=\sqrt{\sum_{m=1}^{128}\left(d_{k,i,m}-d_{k-1,j,m}\right)^2} \tag{4.3}$$

After feature matching step, features in current step will be linked to previous features point-to-point. For a certain feature, its 3D positions (x,y,z) in stereo camera coordinate are also known from stereo measurements. Therefore, each of the features positions in consecutive time steps are known and the camera related pose can be estimated by using these spatial relations.

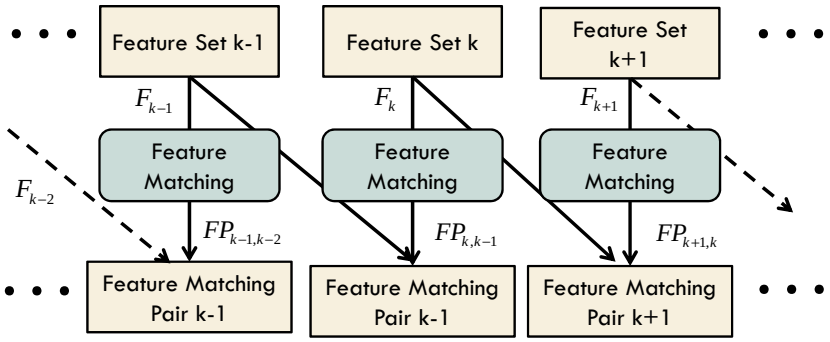
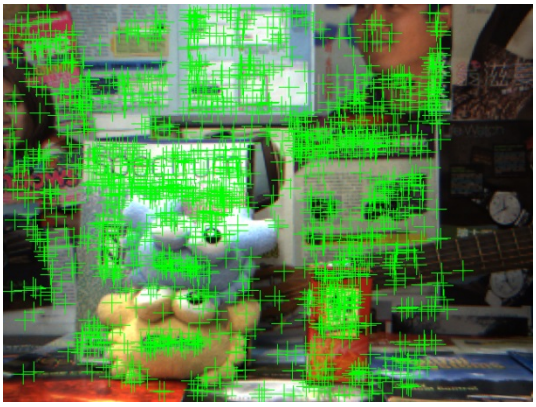
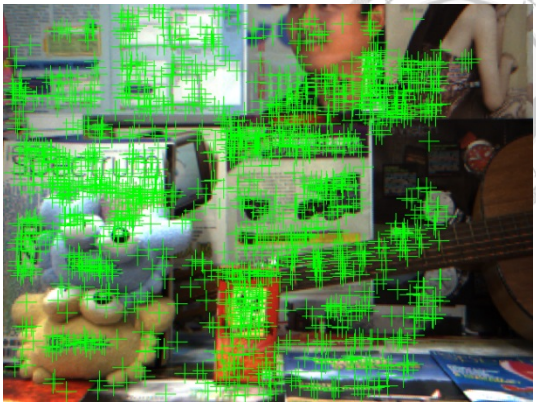


Figure 4.7: Block diagram of feature matching processing



(a)



(b)

(c)

Figure 4.8: The result of feature matching by estimate the similarity between two feature descriptor.

(a) Previous image with the detected features. The same as in Figure 4.6(e).

(b) Current image with the detected features. The same as in Figure 4.6(f).

(c) By comparing the similarity, each previous feature is linked to the current feature as the same landmark.

4.1.3 Estimate the relative transformation matrix of rigid body by Least-Squares method using SVD.

After finishing the feature matching stage, points in each matching pair is assumed to be the same landmark in the environment with different positions at current and previous frames in camera coordinate. The goal is to find a proper rotation and translation to fit the current feature set in matching pair to the previous feature set. This task is well known as "point clouds registration." To find the optimal 3×3 rotation matrix R and the 3×1 translation vector t, the least-squares fitting method using singular value decomposition (SVD) is first proposed in [15: Arun et al. 1987]. To simplify the problem, two point sets are assumed to be a rigid body. Each point in rigid body has the same rotation and translation in an arbitrary motion with respect to its centroid, which is formulated as follows.

$$PFP_{k-1} = R \times PFP_k + t \tag{4.4}$$

Therefore the rotation and translation can be departed in two independent materials. According to the work proposed by [15: Arun et al. 1987], the overall least-square fitting algorithm is listed in Algorithm 4.1, and the concept is illustrated in Figure 4.9. Figure 4.9(a) shows that a rigid body is captured by camera in $(k-1)$-th and k-th steps, where rigid body is located at different positions with respect to different camera coordinates, as shown in Figure 4.9(b) and Figure 4.9(c). Figure 4.9(d) shows that these

two measurements are drawn in the same coordinate. The dot lines in Figure 4.9(d) stand for the measurement in $(k-1)$-th step, while the solid lines stand for the measurements in k-th step. To align these measurement points, the rotation part R needs to be estimated before the translation since the translation part depends on $PFP_k \times R$. Because each point of the rigid body rotates around its centroid [60: Spong 2005], the rotation can be estimated by ignoring the translation part by setting two centroids to the original, as shown in Figure 4.9(f). Therefore the centroids of two point sets are calculated at first as in Algorithm 4.1 line 2-3 and each point is subtracted to its centroid in the product terms of 3×3 covariance matrix in Equation (4.5). The optimal rotation can be calculated using the following covariance matrix:

$$H = \sum_{i=1}^{N_{Pairs}} (PFP_{i,k-1} - \overline{X}_{k-1})(PFP_{i,k} - \overline{X}_k)^T \tag{4.5}$$

Note that the order of the product in Equation (4.5) cannot be changed or the transformation will be the inverse motion. According to the property of SVD, the covariance H can be decomposed into three parts, as shown in Equation (4.6), where H is the product of these parts and is written as follows.

$$[U,S,V] = SVD(H) \tag{4.6}$$

$$H = USV^T \tag{4.7}$$

Therefore, the optimal least-squares rotation matrix can then be calculated as follows:

$$R = VU^T \tag{4.8}$$

After the optimal rotation part is calculated, the translation term can be extracted. As the properties of rigid body mention above, each point has the same translation as its centroid. Therefore, the corresponding optimal translation vector can be calculated as follows:

$$t = \overline{X}_{k-1} - R\overline{X}_k \tag{4.9}$$

The transformation matrix can be written as a 4×4 matrix combining with rotation matrix and translation vector together into homogeneous transformation form, which is written as follows [60: Spong 2005]:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{k,k-1} & T_{k,k-1} \\ 0 & 1 \end{bmatrix} = M_{k,k-1} \tag{4.10}$$

Algorithm 4.1: Estimate relative transformation matrix using SVD
Input: Feature matching pairs $FP_{k,k-1}$.
Output: 4×4 transformation matrix $M_{k,k-1}$
1: Calculate the number of point pairs $N_{Pairs} = \text{size}(FP_{k,k-1})$;
2: Compute the centroid of point set in step k, $\overline{X}_k = (\bar{x}_k, \bar{y}_k, \bar{z}_k)^T = \frac{1}{N_{Pairs}} \sum_{i=1}^{N_{Pairs}} PFP_{i,k}$
3: Compute the centroid of point set in step k-1, $\overline{X}_{k-1} = (\bar{x}_{k-1}, \bar{y}_{k-1}, \bar{z}_{k-1})^T = \frac{1}{N_{Pairs}} \sum_{i=1}^{N_{Pairs}} PFP_{i,k-1}$
4: Compute the covariance matrix $H = \sum_{i=1}^{N_{Pairs}} (PFP_{i,k-1} - \overline{X}_{k-1})(PFP_{i,k} - \overline{X}_k)^T$
5: Decompose covariance matrix H by SVD, $[U,S,V] = SVD(H)$
6: Calculate rotation matrix $R = VU^T$
7: Calculate translation vector $t = \overline{X}_{k-1} - R\overline{X}_k$
8: Combine rotation and translation together. $M_{k,k-1} = [R,t;0,1]$;

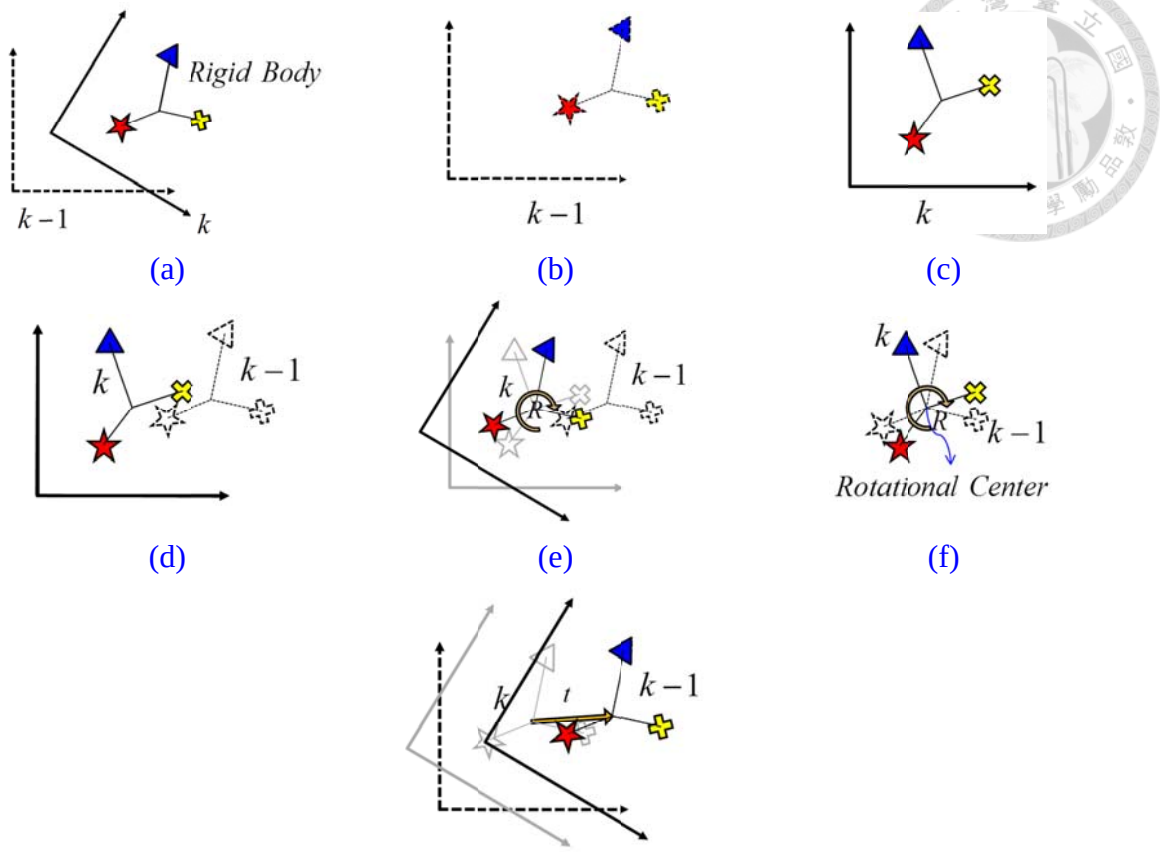


Figure 4.9: Illustration of two point sets with a certain motion.

(a) Capturing a rigid body with different camera poses with respect to the consecutive from $(k-1)$-th to k-th step.

(b) Camera measurement in $(k-1)$-th step.

(c) Camera measurement in k-th step.

(d) Put these two measurements together in the same camera coordinate

(e) Find the rotation matrix by comparing the current data points to previous data points

(f) Since each points on rigid body have the same rotation angle according to its centroids, two measurements on the same rigid body in $(k-1)$-th and k-th step will have the same rotation centroids. This figure illustrates the concept that the rotation matrix R is estimated by ignoring the translation and rotating each point in $(k-1)$-th to k-th step in order to find the least-square rotation matrix.

(g) After finishing estimatin the rotation matrix, the translation vector is then estimated by subtracting the k-th step centroid position to the rotated $(k-1)$-th step centroid position.

Equation (4.10) is the relative camera motion from $(k-1)$-th to k-th step. In order to place stereo measurements to global coordinate, camera pose needs to be specified. The relation between camera relative motion and the camera pose in each step is illustrated in Figure 4.10. The camera pose in k-th step can be defined as C_k and the pose in $(k-1)$-th step can be defined as C_{k-1}. Therefore, the relation between C_k and C_{k-1} can be written as follows:

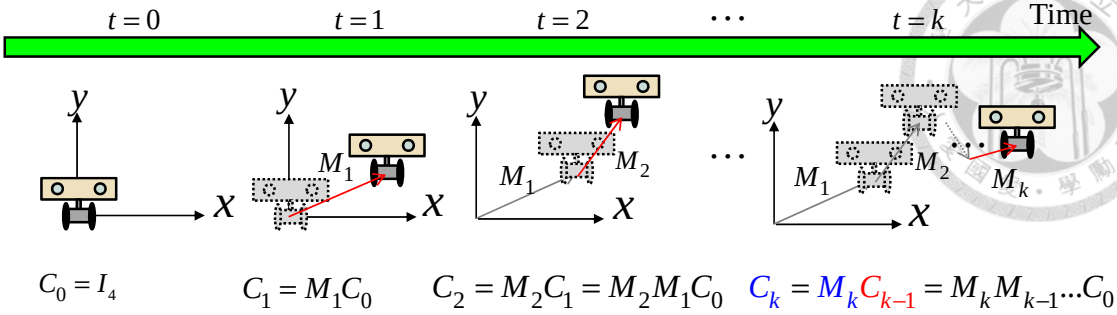
$$C_k = M_{k,k-1} C_{k-1} \tag{4.11}$$

Similarly, the camera at $(k-1)$-th step C_{k-1} can be written as $C_{k-1} = M_{k-1,k-2} C_{k-2}$, and the pose at k-th step can then be written as $C_k = M_{k,k-1} C_{k-1} = M_{k,k-1} M_{k-1,k-2} C_{k-2}$. Following this rule, if the initial pose C_0 is known or predefined by an identity matrix as global coordinate, the camera pose at k-th step can be written as follows:

$$C_k = M_{k,k-1} M_{k-1,k-2} \ldots M_{1,0} C_0 = \left(\prod_{i=k}^{1} M_{k,k-1} \right) C_0 \tag{4.12}$$

Each k-th step point can be transformed to initial step camera coordinate as the defined global coordinate, and this relation can be written as follows:

$$\begin{bmatrix} X_k^{Global} \\ Y_k^{Global} \\ Z_k^{Global} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ Z_k \\ 1 \end{bmatrix} \Rightarrow P_k^{Global} = C_k P_k \tag{4.13}$$



$C_0 = I_4$ $C_1 = M_1 C_0$ $C_2 = M_2 C_1 = M_2 M_1 C_0$ $C_k = M_k C_{k-1} = M_k M_{k-1} ... C_0$

Figure 4.10: Illustration of the relation between camera pose and relative camera motion.

4.1.4 Camera Pose Estimation with RANSAC Outlier Rejection

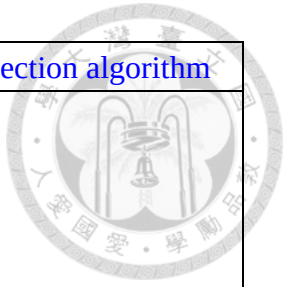
In previous section, the distinctive feature points are detected in each step, and the corresponding feature points will be matched in two consecutive frame data. The camera relative pose can then be estimated by using the spatial relation between these matching pairs using SVD method. However, using these matching pairs without any selection will cause inaccurate or incorrect localization result. Although the SIFT feature is quite robust comparing to most of the recent feature techniques, there might have some wrong matching cases such as repeating features or similar object in the world. In addition, the uncertainty of each stereo camera measurement point may contribute some drift to the final relative pose. To overcome the above problem, Random Same Consensus (RANSAC) outlier rejection framework is applied to find a best transformation matrix. The modified RANSAC algorithm to this case is listed in

Algorithm 4.2. It is assuming that the best transformation matrix is the model with largest number of inliers. In each iteration step, several matching pairs are selected randomly as sample data to estimate a transformation matrix $M_{current}$ by SVD (as in Algorithm 4.1). In order to determine which matching pairs are inliers, the feature points in current frame are transformed by $M_{current}$ to previous camera coordinate, then each spatial error of the matching pair can be calculated by using Euclidean distance, which can be written as:

$$\varepsilon = d(PFP_{i,k}^{*}, PFP_{i,k-1}) = \sqrt{\left(x_{i,k}^{*} - x_{i,k-1}\right)^2 + \left(y_{i,k}^{*} - y_{i,k-1}\right)^2 + \left(z_{i,k}^{*} - z_{i,k-1}\right)^2} \quad (4.14)$$

$$\begin{bmatrix} PFP_{i,k}^{*} \\ 1 \end{bmatrix} = M_{current} \begin{bmatrix} PFP_{i,k} \\ 1 \end{bmatrix} \quad (4.15)$$

where $PFP_{i,k}^{*} = (x_{i,k}^{*}, y_{i,k}^{*}, z_{i,k}^{*})^T$ is the coordinate of $PFP_{i,k}$ which is transformed by $M_{current}$ as shown in Equation (4.15). If the Euclidean distance between $PFP_{i,k}^{*}$ and $PFP_{i,k-1}$ is less than a predefine threshold, it is considered to be an inlier. By doing so, the transformation matrix and inlier set are calculated in certain step. Then, after $N_{Iteration}$ times iteration, there will be $N_{Iteration}$ number of transformation matrix M_i and the corresponding inlier sets $Inliers_i$, where $i = 0...N_{Iteration}$. The best transformation M_{best} is determined by choosing the transformation matrix M_i with the largest number of inliers $Inliers_i$, which can be done iteratively without storing all the trying models M_i with its inliers $Inliers_i$ as in the line 11-21 in Algorithm 4.2.



Algorithm 4.2: Feature-based localization with RANSAC outlier rejection algorithm

Input:

Feature Matching Pair $FP_{k,k-1}$

Set the number of iterations $N_{Iteration}$

Number of sample pairs N_{sample}

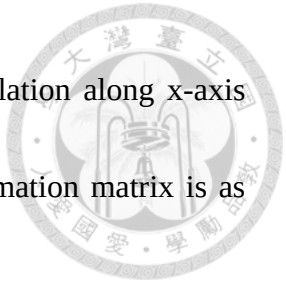
Output:

Transformation Matrix $M_{t,t-1}$

Inlier list $Inliers$

1: Initialize best transformation matrix $M_{best} \leftarrow \phi$
2: Initialize best inliers set $Inliers_{best} \leftarrow \phi$
3: Initialize best inliers number $N_{Inliers\_best} = 0$
4: Calculate the number of matching pairs $N_{Match} = size(FP_{k,k-1})$
5: ***for*** iteration = 1: $N_{Iteration}$
6: $SampleSet \leftarrow$Randomly select N_{sample} matching pairs in $FP_{k,k-1}$
7: Compute Current Transform Matrix $M_{current}$ from $SampleSet$ using SVD (Algorithm 4.1)
8: $Inliers_{Current} \leftarrow \phi$
9: ***for all*** $FP_{k,k-1,i}$ in $FP_{k,k-1}$
10: Compute the spatial error between $M_{current}$ $PFP_{i,k}$ and $PFP_{i,k-1}$ by using Euclidean distance, that is, $\varepsilon = Euclidean(M_{current}PFP_{i,k}, PFP_{i,k-1})$
11: ***if*** $\varepsilon < threshold$
12: $Inliers_{Current} \leftarrow Inliers_{Current} + FP_{k,k-1,i}$
13: ***end if***
14: ***end for***
15: Count the number of $Inliers_{Current}$, $N_{Inliers\_current} = size(Inliers_{Current})$
16: Recomputing the transformation matrix $M_{current}$ by $Inliers_{Current}$ using SVD (Algorithm 4.1)
17: ***if*** $N_{Inliers\_current} > N_{Inliers\_best}$
18: $M_{best} \leftarrow M_{current}$
19: $Inliers_{best} \leftarrow Inliers_{Current}$
20: $N_{Inliers\_best} = N_{Inliers\_current}$
21: ***end if***
22: ***end for***
23: $M_{k,k-1} \leftarrow M_{best}$

For better understanding, 2-th and 3-th frame data are taken as $(k-1)$-th and



k-th steps for example. The given relative motion is a pure translation along x-axis with positive 0.1m without any rotation, and therefore the transformation matrix is as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.16}$$

Figure 4.11 (a)-(b) are two target images captured from right CCD of the stereo camera with corresponding feature points in $(k-1)$-th and k-th steps, respectively. The green circles indicate the features in the $(k-1)$-th step, while the red dots represent the features in the k-th step. Figure 4.11(c) and (d) show the projecting result of k-th step features from 3D coordinate to image plane by pin-hole model with different transformation matrixes estimated in two iterations. For better estimation iteration case, features in k-th step are transformed by the following matrix:

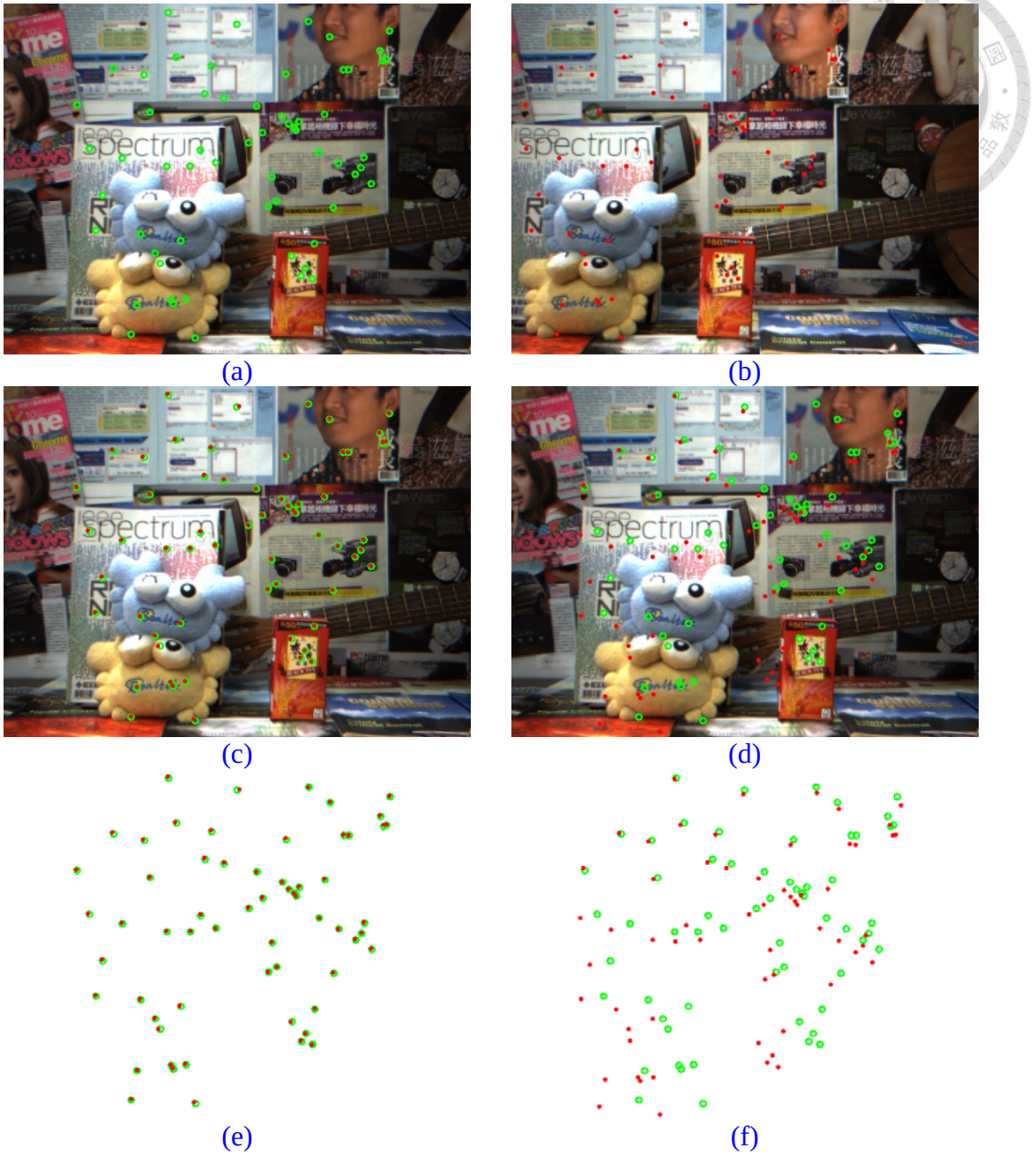
$$T = \begin{bmatrix} 1.0000 & -0.0019 & -0.0033 & 0.1019 \\ 0.0019 & 1.0000 & -0.0002 & 0.0009 \\ 0.0033 & 0.0002 & 1.0000 & 0.0008 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.17}$$

Most of the red dots align to the green circles as shown in Figure 4.11(c) and (e). The aligning pairs are equivalence to the 3D spatial inliers since the projection by camera pin-hole model is a degeneration process. This means that each feature in k-th step in the matching pairs is transformed correctly to the corresponding feature in $(k-1)$-th step. On the other hand, the incorrect transformation matrix is estimated in the second iteration case, and most of the red dots do not align to the green circles in the result of

features projection shown in Figure 4.11(d) and (f). The corresponding transformation matrix in the second iteration is as follows:

$$T = \begin{bmatrix} 0.9933 & -0.0536 & 0.1021 & -0.0177 \\ 0.0584 & 0.9973 & -0.0441 & 0.0605 \\ -0.0995 & 0.0498 & 0.9938 & -0.0068 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

The number of the inliers in the second iteration is less than the number of the inliers in the first iteration dramatically, and therefore this example shows that the relation between the best transformation matrix and the number of the corresponding inliers. The final result of RANSAC outlier rejection algorithm is shown in Figure 4.12(b), only the green lines are considered to be the inputs of the camera pose estimation step, whereas the red lines are the outliers and do not be considered into the pose estimation step.



(a) (b) (c) (d) (e) (f)

Figure 4.11: Illustration of estimating the relative motion with RANSAC algorithm by two iterations for example. Green circles indicate the feature points in $(k-1)$-th step, while the red dots indicate the feature points in k-th step. Feature points in k-th step are projected by pin-hole camera model with certain transformation matrix.

(a)-(b) Feature points in $(k-1)$-th and k-th steps respectively.

(c) Feature points projected with correct transformation matrix.

(d) Feature points projected with incorrect transformation matrix. It is obvious that a lot of red dots transformed by incorrect transformation matrix do not align to the green circles.

(e)-(f) The corresponding features plotting without showing images for better visualization to distinguish these points.

(a)

(b)

Figure 4.12: The result of using RANSAC outlier rejection algorithm on the matching pairs.

(a) By comparing the similarity, each previous feature is linked to the current feature as the same landmark.

(b) With RANSAC outlier rejection, some wrong matching pairs are removed. Green lines indicate the inliers, whereas the red lines represnet the outliers that do not consider into the motion estimation process.

4.2 Stereo Vision Refinement

In this section, a simple way to refine the disparity image is presented. For window-based stereo process, due to texture-less, lighting problem and occlusion case in capturing environment, there might have many missing data (broken holes) in disparity map. Therefore, these missing data regions are detected and filled by the proposed method. Connected-component labeling is used to recognize and analyze each region. These broken regions can be filled by the proposed interpolation or radial basis function. These two interpolation methods are discussed in Subsection 4.2.1 and compared in Subsection 6.2.7.

Moreover, some parts in the target image that cannot be seen by the reference image plane. These parts should not have measurements. However, in practice this region has some measurements due to the mismatching of using the window-based stereo reconstruction method. In this thesis, these parts are defined as forbidden area, and measurements in this area are removed by the proposed method.

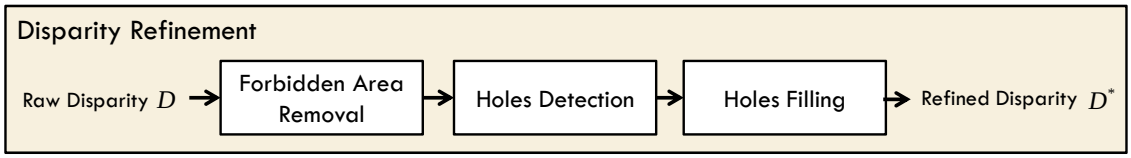


Figure 4.13: The block diagram of the proposed stereo refinement algorithm

4.2.1 Forbidden Area Detection and Elimination

In this section, the proposed stochastic-based disparity forbidden area detection and removing method is discussed. One of the limitations of two lens stereo camera rig is that there has a region that can be seen by target image but another image plane. Figure 4.14(a) is a typical stereo camera configuration, while the target image is defined as the right image plane. The red dash line at the right side of the right camera indicates the region that can be seen by target image plane but cannot be seen by left image plane. This area cannot find corresponding feature patch from right to left image, as shown in Figure 4.14(b), and therefore it should not have any measurement data in that region. Figure 4.14(c) is the disparity map corresponding to the target image as the right image in Figure 4.14(b). The region colored in translucent purple region in the right side of the disparity map indicate the region that should not have any measurement. However, due to the limitation of window-based matching technique which uses sum of average difference (SAD) as similarity index, many wrong matching points cause wrong measurements at that region, which is defined as “forbidden area” in this thesis. Moreover, the size of the forbidden area varies with the distance from objects to camera. The larger of the distance between object to camera, the boundary of the forbidden area is closer to the right side of image plane (and thus the smaller region), as illustrated in Figure 4.14(d). In most cases, the forbidden region is filled with invalid pixels in the

right side of the disparity map. Therefore the statistical method proposed in this thesis is used to solve this problem. The block diagram of the proposed forbidden area detection and removing method is shown in Figure 4.15.

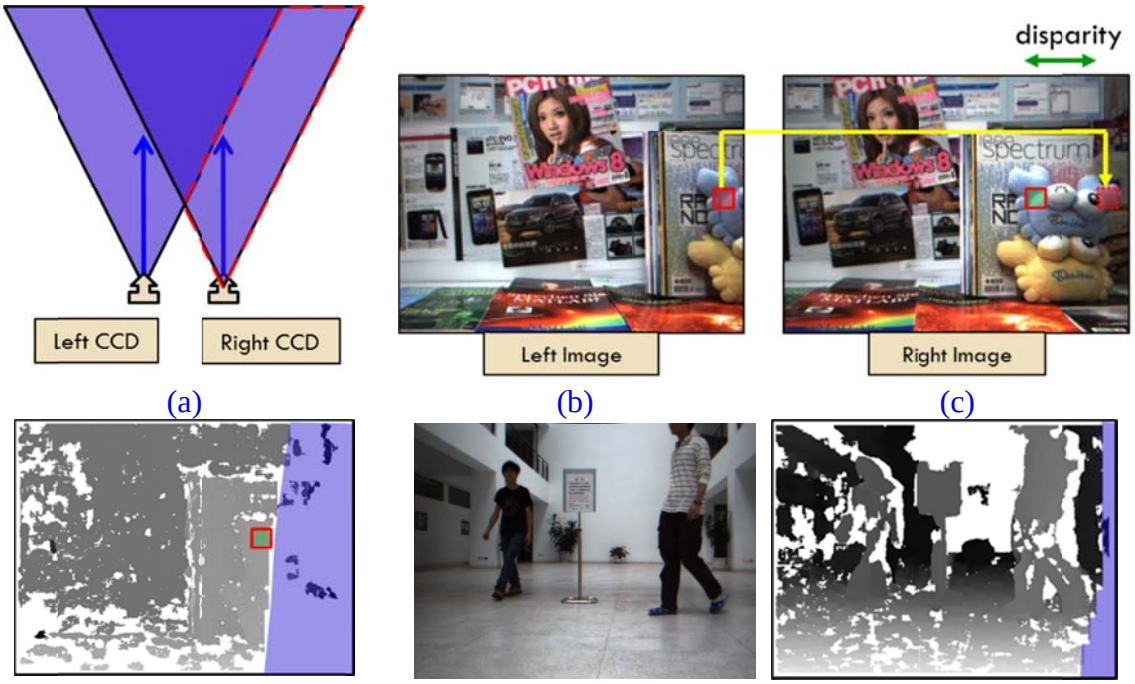


Figure 4.14: Illustration of the occlusion area that out of the field of view (FOV) of left image plane

(a) The geometry configuration of two camera planes of the stereo rig. The red line area can be seen by right image plane but cannot by left image plane.

(b)-(c) Two images retrieved from left and right image plane. Some parts at the right side of the right image that cannot be found out in the left image. These pixels in the right image should not have disparity since there cannot find a corresponding point in the left image.

(d) Disparity map corresponding to the target image which is defined to be the right image in this thesis. The right side of the disparity colored in translucent purple should not have value in ideal. However, in practice there have some wrong measurements in that region.

(e)-(f) For sensor measures in farther distance environment, the effect of this phenomenon is less than the closer measurement object.

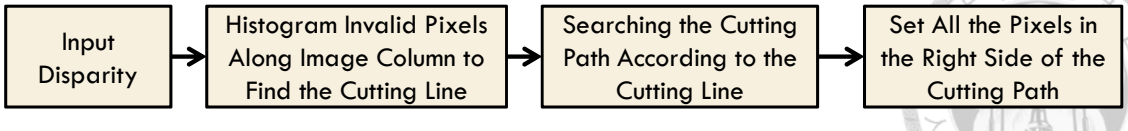


Figure 4.15: The block diagram of forbidden area detection

For an arbitrary target image with corresponding disparity map shown in Figure 4.17(a) and (b), the disparity map is accumulated on the invalid pixels along the image column at first. That is, the accumulator counts the number of all invalid pixels in j-th column. It can be formulated as follows:

$$A_{I_d}(j) = \sum_{i=1}^{ImageRow} B\left(I_d(i,j)\right) \tag{4.19}$$

where

$$B\left(I_d(i,j)\right) = \begin{cases} 1, & if\ I_d(i,j) = 256 \\ 0, & other \end{cases} \tag{4.20}$$

A_{I_d} is a $1 \times ImageCol$ vector which stores the number of invalid pixels in disparity map. Since some sparkle signals occur in A_{I_d}, the average filter is applied to A_{I_d} before searching the cutting line. It can be written as follows:

$$A^*_{I_d}(j) = \frac{1}{N} \sum_{i=j-N/2}^{j+N/2} A_{I_d}(i) \tag{4.21}$$

To find a reasonable cutting line $C_{Cutting}$ on I_d to depart the forbidden area, the maximum gradient in $A^*_{I_d}(j)$ is found according to the assumption that the right side of the boundary of forbidden area has large number of invalid pixels while the left side has large number of valid pixels. This can be written as follows:

$$C_{Cutting} = \underset{j=ImageCol/2:ImageCol}{\arg\max} \left(A^*_{I_d}(j) - A^*_{I_d}(j-1) \right) \tag{4.22}$$

The cutting line extraction process is listed in Algorithm 4.3. Figure 4.17(c) shows the

accumulation result on invalid pixels along the column in the disparity map, and Figure 4.17(d) shows the resulting cutting line plotting on the disparity map. This process departs the disparity map into the reasonable and the forbidden areas roughly by the statistical vertical cutting line.

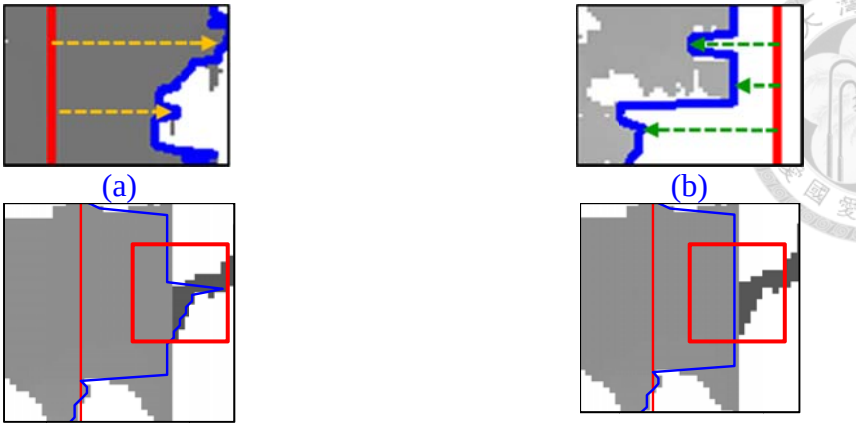
Algorithm 4.3: Cutting line extraction
Input: Disparity map I_d
Output: Cutting line $C_{Cutting}$ that stores the u-coordinate.
1: Find out the disparity map size $[ImageRow, ImageCol] = \text{size}(I_d)$;
2: Initialize the accumulator, $Accumulator\ A_{I_d} = zeros(1, ImageCol)$;
3: $C_{Cutting} = ImageCol,\ \varepsilon_{G,Max} = 0$
4: ***for*** $j = 1$ *to* $ImageCol$
5: $A_{I_d}(j) = sum(I_d(:, j) == 256)$;
6: ***end for***
7: $A_{I_d} = Average\_Filter(A_{I_d})$
8: ***for*** $j = ImageCol$ *to* $ImageCol / 2$
9: $\varepsilon_{G,Current} = A_{I_d}(j) - A_{I_d}(j-1)$
10: ***if*** $\varepsilon_{G,Current} > \varepsilon_{G,Max}$
11: $C_{Cutting} = j - 1$
12: $\varepsilon_{G,Max} = \varepsilon_{G,Current}$
13: ***end if***
14: ***end for***

Most of invalid pixels are on the right side of the cutting line and are eliminated in the next step. However, some valid pixels will be removed by this rough constraint. Therefore, the cutting path $P_{Cutting}$ is found according to the position of the cutting line to avoid this problem. The cutting path is a $ImageRow \times 1$ vector and is searched from top to bottom of image plane. To find the cutting path, it follows the basic concepts: if a pixel located on the cutting line is valid, the cutting path is found by searching the

boundary of the valid pixel from left to right starting at $C_{Cutting}$, as shown in Figure 4.16(a), or the cutting path is determined by searching the boundary of the invalid pixel from right to left starting at $C_{Cutting}$, as shown in Figure 4.16(b). However, applying this method directly causes false result since there has wrong disparity near the boundary, as shown in Figure 4.16(c). To overcome this problem, the additional continuity constraint is applied during right searching the invalid pixel. The constraint is checking the continuity of path disparity, that is:

$$abs(I_d(i,j) - D_{Path}(i-1)) > \lambda_{Contineous} \tag{4.23}$$

$D_{Path}(i-1)$ is the disparity of the v-coordinate $(i-1)$ on the cutting path, $I_d(i,j)$ is the disparity of the coordinate (i,j), $\lambda_{Contineous}$ is the user define threshold. Since this constraint is set to check if the disparity is close to the previous step, the threshold is set as 0.1 times $D_{Path}(i-1)$, that is, $\lambda_{Contineous} = 0.1 \times D_{Path}(i-1)$. Therefore, a cutting path can be extracted to eliminate the wrong measurements pixels at the right side of the disparity map, as shown in Figure 4.17(e)-(f).



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

Figure 4.16: Illustrate the basic concept of the cutting path extraction according to cutting line.

(a) Right search the invalid pixel as the boundary path.
(b) Left search to find the valid pixel to be the boundary path.
(c) Since wrong measurements are near the object boundary, directly apply the method will cause wrong path result.
(d) The constraint is applied to check the continuity of the disparity of the last step.

Algorithm 4.4: Cutting path extraction

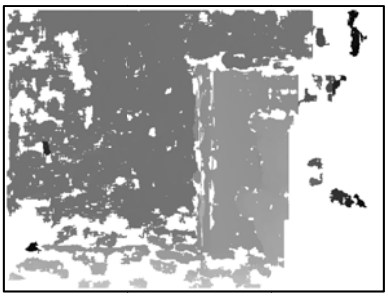
Input: Disparity map I_d
Cutting line $C_{Cutting}$

Output: Cutting path $P_{Cutting}$ vector with length $ImageRow \times 1$ stored each corresponding u-coordinate.

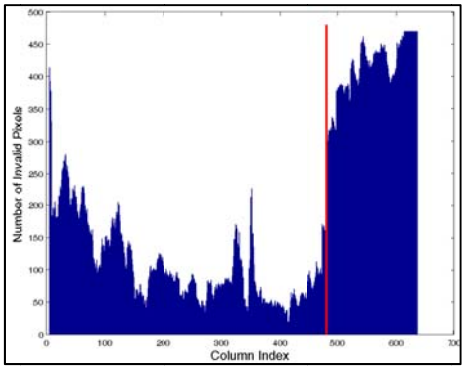
1: Find out the disparity map size $[ImageRow, ImageCol] = \text{size}(I_d)$;
2: Initialize the cutting path, $P_{Cutting} = ones(imageRow,1) \times C_{Cutting}$
3: Initialize the path disparity, $D_{Path} = ones(imageRow,1) \times (imageCol - C_{Cutting} - 2 \times W_{Edge})$
4: ***for*** $i = W_{Edge} + 1$ *to* $ImageRow - W_{Edge}$
5: $\lambda_{Contineous} = D_{Path}(i-1) \times \tau_{Contineous}$
6: ***if*** $I_d(i, C_{Cutting}) == 'invalid'$ ***or*** $abs(I_d(i, C_{Cutting}) - D_{Path}(i-1)) > D_{Path}(i-1) \times \tau_{DisparityError}$
7: ***for*** $j = W_{Edge} : -1 : ImageCol / 2$ //Left search
8: ***if*** $I_d(i, j) \sim= 'invalid'$ ***and*** $(I_d(i, j) - D_{Path}(i-1)) < \lambda_{Contineous}$
9: $D_{Path}(i) = I_d(i, j)$
10: $P_{Cutting}(i) = j$
11: ***break***
12: ***end if***
13: ***end for***
14: ***else***
15: ***for*** $j = P_{Cutting}(i-1)$ *to* $ImageCol$ //Right search
16: ***if*** $I_d(i, j) == 'invalid'$ ***or*** $abs(I_d(i, j) - D_{Path}(i-1)) > \lambda_{Contineous}$
17: $P_{Cutting}(i) = j - 1$

18: $D_{Path}(i) = I_d(i, j-1)$
19: ***end if***
20: ***end for***
21: ***end if***
22: ***end for***

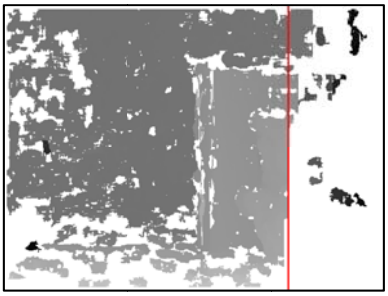
(a)



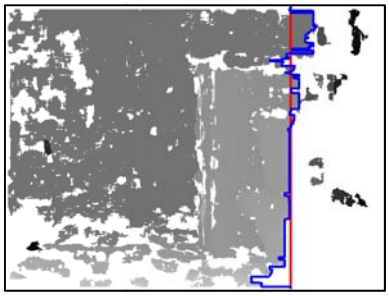
(b)



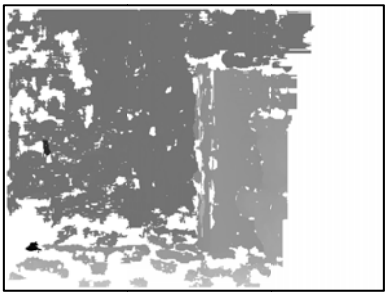
(c)



(d)



(e)



(f)

Figure 4.17: The illustration of the proposed forbidden area detection

(a) The target image captured from right CCD

(b) The raw disparity corresponding to the target image

(c) The result of the accumulating the disparity map along image column and the corresponding cutting line extracted by searching maximum gradient.

(d) The cutting line plotted on the raw disparity map.

(e) The cutting path extracted according to the cutting line.
(f) The resulting disparity with removing pixels in forbidden area.

4.2.2 Holes Detection

The wrong disparity measurements in the forbidden area are removed by the proposed algorithm presented in Section 4.2.1. The proposed missing data area detection algorithm based on connected-component labeling is presented in this section. The block diagram of the proposed holes detection method is shown in Figure 4.18 and illustrated in Figure 4.19. At first, since the value of invalid pixels in disparity map is known (256 in this thesis), these pixels can be marked as '1', and the value of the valid pixels is marked as '0' to form a binary image, as shown in Figure 4.19(d). This can be written as follows:

$$B_{invalid}(i,j)=\begin{cases}1, & \text{if } disparity(i,j)='invalid' \ (=256 \text{ in this theis})\\ 0, & \text{otherwise}\end{cases} \tag{4.24}$$

The holes are detected by applying connected-component labeling process with 4-connectivity mentioned in Section 3.4.4 to the invalid pixels mask $B_{invalid}$, as shown in Figure 4.19(e). Each region R_i which is marked in different colors means the different label. The larger regions are ignored since they are often linked to different objects, as shown in Figure 4.19(f), filling these regions according to different object neighbors will cause wrong result. Larger regions are filtered out by the following equation:

$$Area(R_i) < Th_{Area} \tag{4.25}$$

where $Area(R_i)$ counts the number of pixel in i-th region to be its region area.

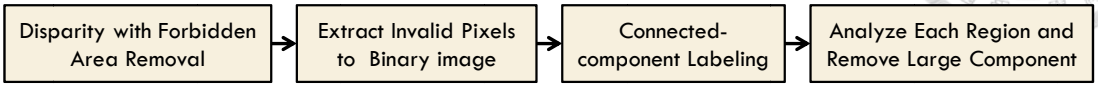


Figure 4.18: The flowchart of holes detection

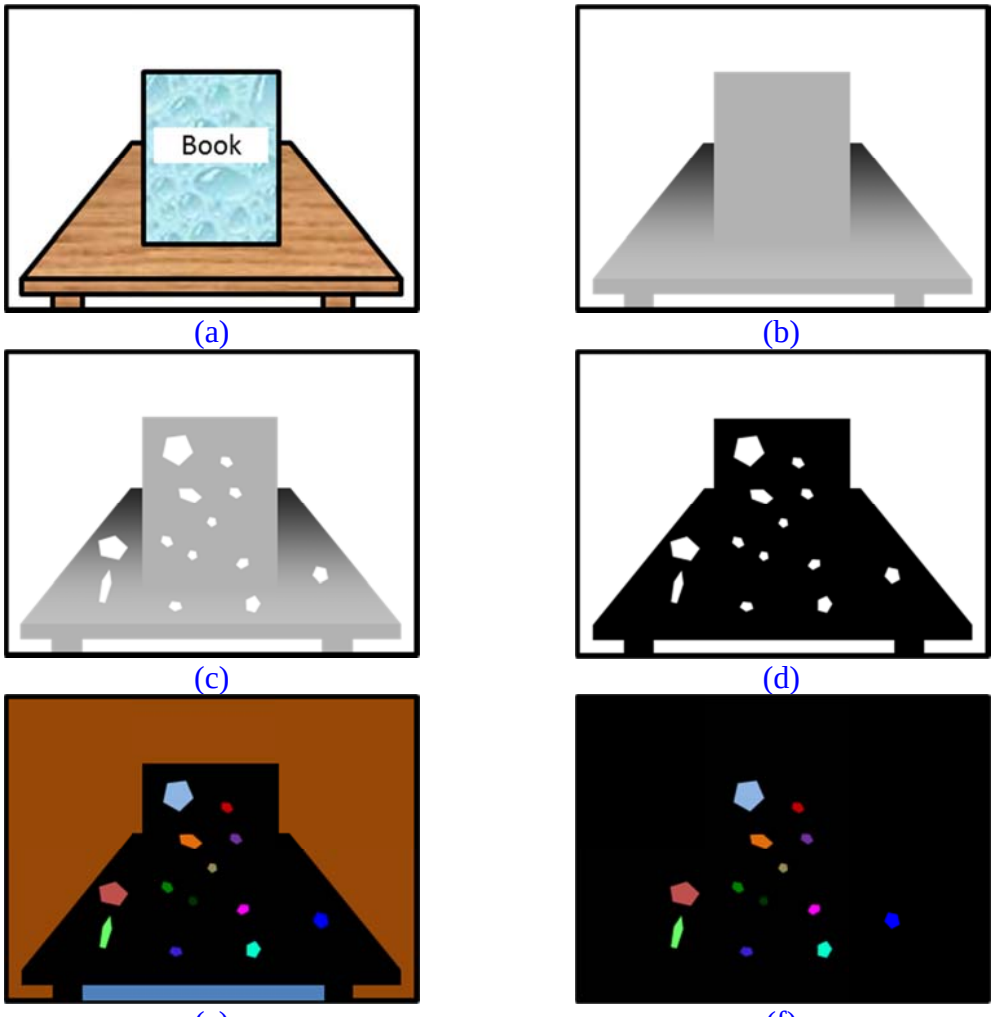


Figure 4.19: Illustration of the hole detection concept

(a) Illustration of right image
(b) Corresponding disparity map of (a) in ideal.
(c) In practice the disparity map have some missing data area.
(d) Invalid mask of disparity map. Pixels colored in white are the missing data.
(e) Using connected-component labeling to identify each region.
(f) The result of removing large area from (e).

4.2.3 Dual Orthogonal Linear Interpolation

The missing data regions are detected by the proposed holes detection method presented in Section 4.2.2. Pixels in these regions are marked as invalid and will be filled by its neighbors. In this section, an interpolation method is proposed to fill the missing data area efficiently. Due to only smaller holes are considered, these regions are assumed to be plane patches. Considering the contributions of horizontal and vertical neighborhood pixels, these patches can be filled by using two linear interpolations on horizontal and vertical directions with same weights, as illustrated in Figure 4.20. For an arbitrary pixel in a broken hole, the pixel can be filled with value d_H by linear interpolation with the horizontal neighbors. The Equation (4.18) shows the horizontal linear interpolation by these two neighbors. u_{Hole} stands for the u-coordinate of certain pixel in the hole. u_L and u_R represent the u-coordinate of the left and right neighbors respectively. d_L and d_R indicate the disparity of the left and right neighbors respectively. On the other hand, the pixel can also be filled with value d_V by linear interpolation with the vertical neighbors, and it can be written as Equation (4.19). v_{Hole} stands for the v-coordinate of the pixel. v_T and v_B represent the v-coordinate of the top and bottom neighbors respectively. d_T and d_B indicate the disparity of the top and bottom neighbors respectively. Assuming d_H and d_V have same contribution, the final disparity value of the pixel d_{Total} can be calculated as the average of vertical

and horizontal interpolations as in Equation (4.20), and the above process is listed in Algorithm 4.5.

$$d_H = \left(\frac{d_R - d_L}{u_R - u_L} \right) (u_{Hole} - u_L) + d_L \tag{4.26}$$

$$d_V = \left(\frac{d_B - d_T}{v_B - v_T} \right) (v_{Hole} - v_T) + d_T \tag{4.27}$$

$$d_{Hole} = d_{Total} = 0.5(d_H + d_V) \tag{4.28}$$

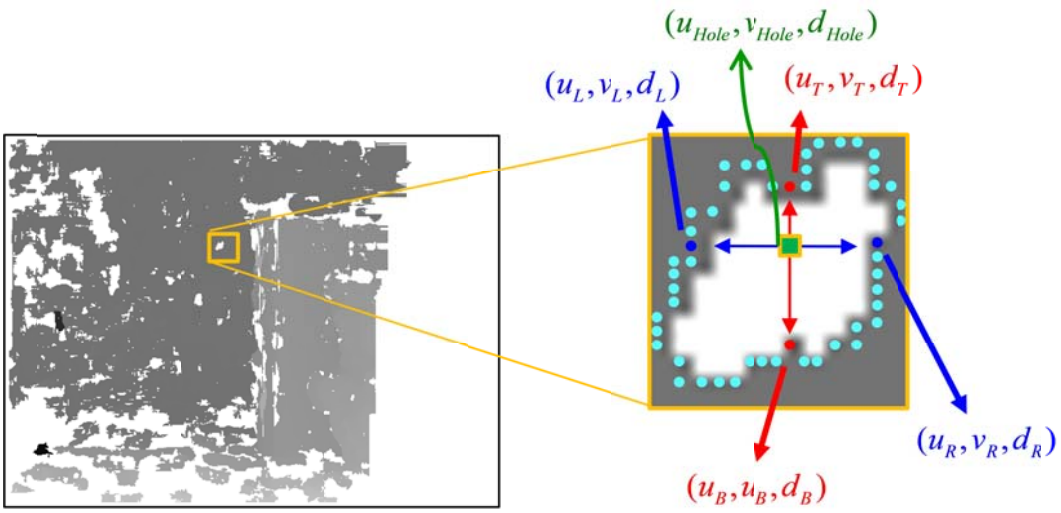
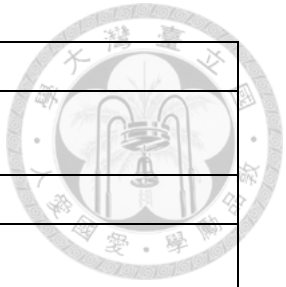


Figure 4.20: Illustration of hole filling on an arbitrary pixel by DOL interpolation. Blue arrows indicate the searching direction in horizontal part. Two neighbors marked as blue dots are selected by searching the right and left side of the pixel. Similarly, the red arrows indicate the searching direction in vertical part, and the corresponding neighbors marked as red dots are chosen by searching the top and bottom side of the pixel. Light blue dots represent the neighbor candidates of the hole.



Algorithm 4.5: Hole filling with DOL interpolation

Input: Disparity Map with forbidden area rejection $I_{d,FR}$
Region properties structure R.

Output: The interpolated disparity map I_d^*

Find out the disparity map size $[ImageRow, ImageCol] = \text{size}(I_d)$;
for all regions R_i in R
  for all pixels P_j in i^{th} region R_i
    for $k = u_{Hole} : -1 : 1$
      if $I_{d,FR}(v_{Hole}, k) == 'valid'$
        $u_L = k,\ d_L = I_{d,FR}(v_{Hole}, k)$
        break
      end if
    end for
    for $k = u_{Hole} : ImageCol$
      if $I_{d,FR}(v_{Hole}, k) == 'valid'$
        $u_R = k,\ d_R = I_{d,FR}(v_{Hole}, k)$
        break
      end if
    end for
    for $m = v_{Hole} : -1 : 1$
      if $I_{d,FR}(m, u_{Hole}) == 'valid'$
        $v_T = m,\ d_T = I_{d,FR}(m, u_{Hole})$
        break
      end if
    end for
    for $m = v_{Hole} : ImageRow$
      if $I_{d,FR}(v_{Hole}, m) == 'valid'$
        $v_B = m,\ d_B = I_{d,FR}(m, u_{Hole})$
      end if
    end for
    $d_H = (d_R - d_L)(u_{Hole} - u_L)/(u_R - u_L) + d_L,\quad d_V = (d_B - d_T)(v_{Hole} - v_T)/(v_B - v_T) + d_T$
    $I_d^* = 0.5(d_H + d_V)$
  end for
end for

4.2.4 Radial Basis Function

Another data filling method is radial basis function (RBF) which is presented in Section 3.5. RBF considers all the known data with corresponding weights to interpolate the unknown data with the corresponding positions. In order to fill invalid pixels in a hole, neighbors around the hole need to be selected. To obtain the neighbor pixels, the bounding box vertexes of the hole are extracted first. Since the pixels in the hole region are labeled, the smallest bounding box can be easily obtained by finding the minimum and maximum on v- and u-coordinate, as illustrated in Figure 4.21(a) and (b). After extracting the smallest bounding box with four coordinates, $(v_{\min}, u_{\min})$, $(v_{\min}, u_{\max})$, $(v_{\max}, u_{\min})$ and $(v_{\max}, u_{\max})$, the neighbor pixels can be obtained by extracting the valid disparities in the bounding box, as illustrated in Figure 4.21(c) and (d). These neighbors have data structure of $n_i = (u_i, v_i, d_i)$ and are used to be the inputs of RBF.

According to the definition of radial basis function, a hole surface is defined as:

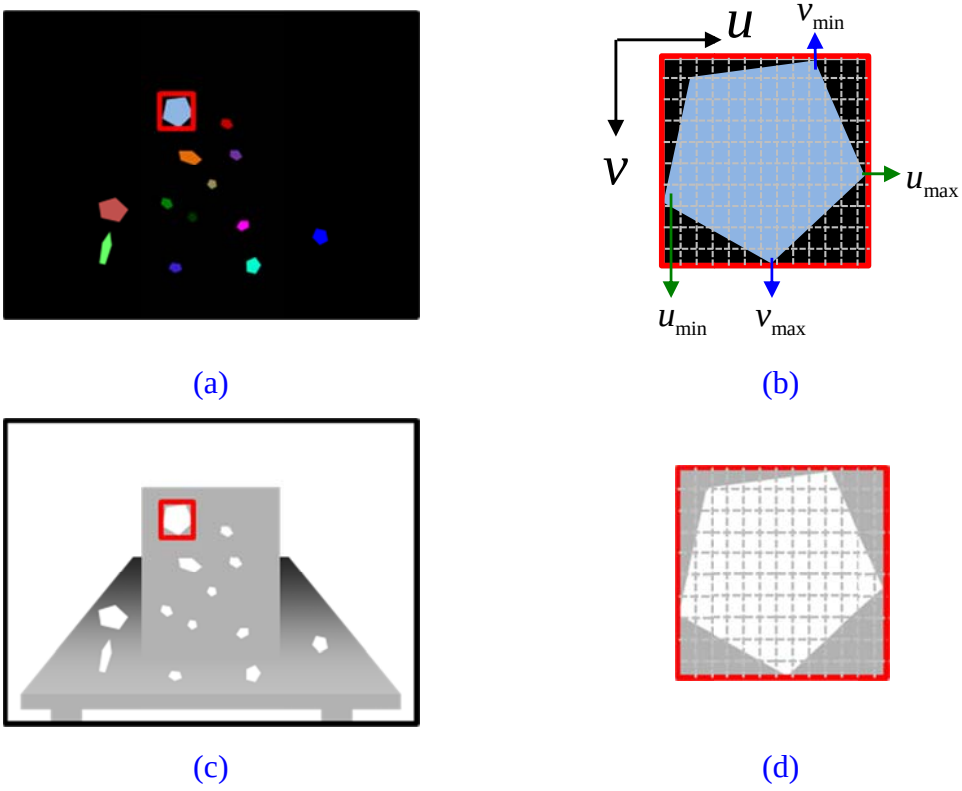
$$y(x) = \sum_{i=1}^{N} \omega_i \cdot \varphi(\|x - x_i\|) \tag{4.29}$$

In this thesis, the multiquadric type is applied and the parameter ε is set to 1. Therefore, substituting ε into the Equation (3.16) the radial function becomes:

$$\phi(r) = \sqrt{1 + r^2} \tag{4.30}$$

where r is the Euclidean distance between neighbor and the processing pixel, that is,

$r = \|x - x_i\|$. x denotes the position of the processing pixel, $(u_{ProcessPixel}, v_{ProcessPixel})$, whereas x_i denotes the position of neighbors, (u_i, v_i). ω_i is the corresponding weights to be determined by known neighbors $n_i = (u_i, v_i, d_i)$ in learning step, as mentioned in Section 3.4. After finishing the learning step, invalid pixel can be interpolated by Equation (4.29).



(a) (b)

(c) (d)

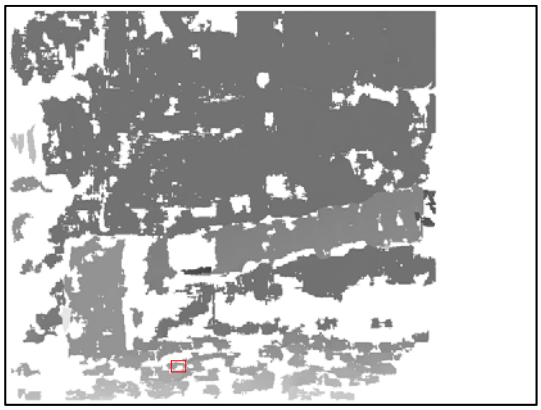
Figure 4.21: Illustration of the smallest bounding box extraction of a certain hole.

(a) Finding the bounding box of the certain hole depicted as red rectangle.

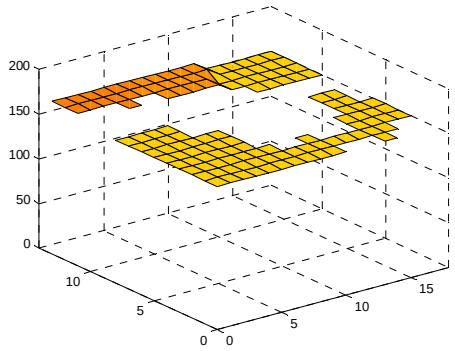
(b) To achieve the goal, pixels belong to that region are checked to find the minimum and maximum of the v- and u-coordinate.

(c) Extracting neighbors in the disparity map in range of the bounding box.

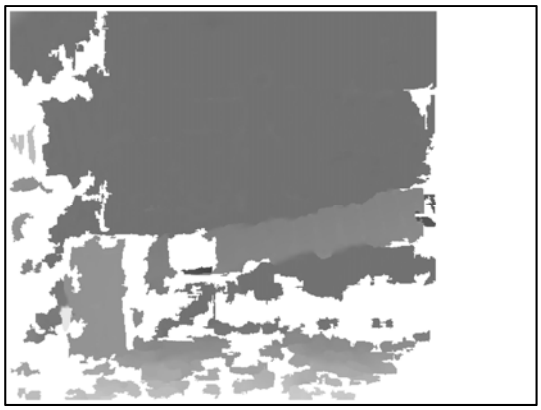
(d) The gray points indicate the valid pixels in disparity map to be the inputs of the RBF available data.



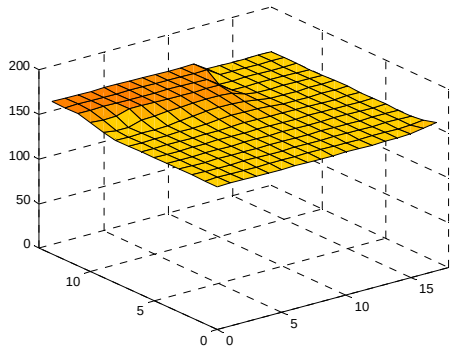
(a)



(b)



(c)



(d)

Figure 4.22: An example of filling a certain hole using radial basis function interpolation.

(a) Disparity map before RBF interpolation. Red rectangle depicts the bounding box of the hole.

(b) 3D mesh plot of the valid neighbors in the bounding box. These neighbors are used to be the input of the RBF as the available known data.

(c) The result of the disparity map after finishing the radial basis function interpolation process.

(d) The surface reconstructed by multiquadric type of RBF using the neighbors data from (b).

Chapter 5

Object Detection and Tracking based on Stereovision

Three-dimensional environment reconstruction system in static scenario is presented in Chapter 4, including sensor localization based on image feature and stereo data preprocessing. However, in real scenarios many dynamic objects affect the localization result due to the localization method uses features as static landmarks. Moreover, the dynamic objects may be seen twice or more in different time step, and are mapped into the 3D model many times that cause ghost effect. For these reason, this thesis proposes the object tracking system for checking the moving object.

In this chapter, the details of the proposed object detection and tracking system are discussed. The overall system architecture is presented in Section 5.1. Section 5.2 presents the proposed object detection algorithm based on u-occupancy grid. Section 5.3 shows the proposed object tracking method using color-based feature with Kalman filter.

5.1 System Architecture

The proposed system can be divided into three parts: stereo data acquisition, object detection and object tracking, illustrated in Figure 5.1. The stereo data acquisition part

provides two color images from stereo camera and the disparity map is calculated by these two images. The object detection and tracking are the main parts of this chapter. The procedures of the object detection part are presented briefly as follows. First of all, the disparity map is transformed to u-disparity occupancy grid map to know where the object is located. Secondly, for the purpose of using image information to analyze the object candidate, object bounding box in image plane needs to be extracted. After each candidate bonding box is extracted, the color information is transformed from RGB to HSV color space. To describe each object by using the color information, two feature vectors are formed by histogramming the pixel value in H and S channels. Therefore, database objects can be registered to the candidates by comparing the correlation of two feature vectors. Finally, a strategy is proposed to handle the update problem.

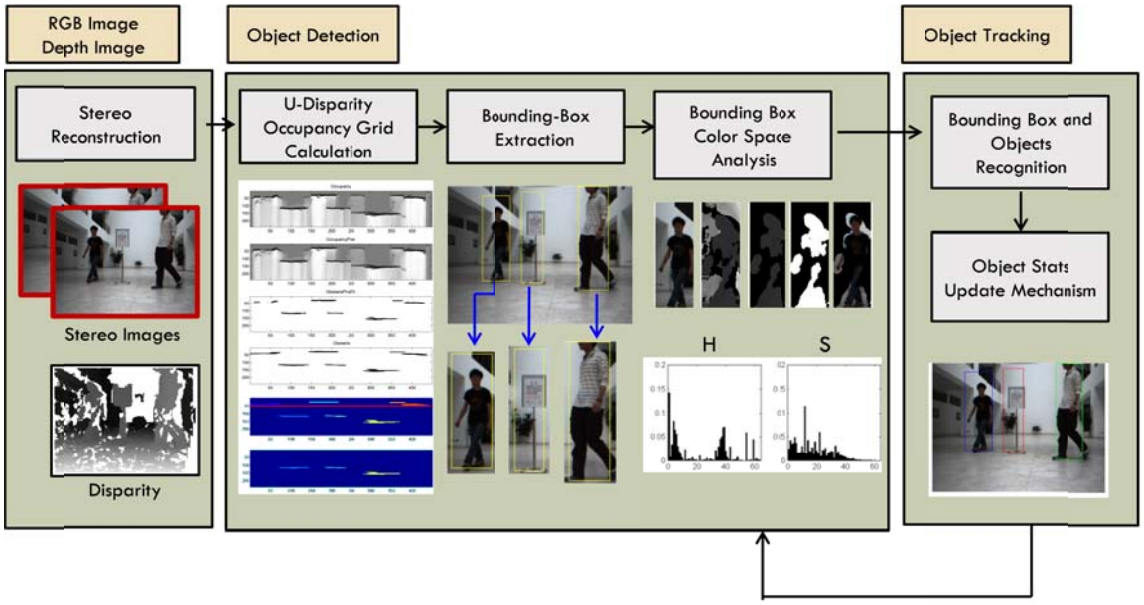


Figure 5.1: The proposed object detection and tracking system

5.2 Object Detection

For describing an object in database easily, some notations are defined in Table 5.1.

Table 5.1: The definition of some notations for a database object

Object Properties	
$\boldsymbol{x}_{i,k}$	Object position in current time step $\boldsymbol{x}_{i,k} = (x_{i,k}, z_{i,k})^T$.
$\boldsymbol{v}_{i,k}$	Object velocity in current time step $\boldsymbol{v}_{i,k} = (v_{i,k,x}, v_{i,k,z})^T$.
$\boldsymbol{m}_{j,k}$	j-th candidate measurement in current time step $\boldsymbol{m}_{j,k} = \left(m_{x,j,k}, m_{z,j,k}\right)$.
$\mathbf{H}_S(O_i)$	The histogram of saturation channel of i-th object.
$\mathbf{H}_H(O_i)$	The histogram of hue channel of i-th object.
$\mathbf{H}_S(C_j)$	The histogram of saturation channel of j-th candidate.
$\mathbf{H}_H(C_j)$	The histogram of hue channel of j-th candidate.
$P_H(O_i)$	The probability distribution of hue channel of i-th object.
$P_S(O_i)$	The probability distribution of saturation channel of i-th object.
$P_H(C_j)$	The probability distribution of hue channel of j-th candidate.
$P_S(C_j)$	The probability distribution of saturation channel of j-th candidate.
State Flags and counters	
$F_{i,InImage}$	Flag that indicates the i-th object is in the field of view of the camera
$F_{i,Occlude}$	Flag that states the i-th object is occluded.
$F_{i,Measurement}$	Flag records if the i-th object is measured by stereo camera.
$Cnt_{i,FOV}$	Counter for accumulating the times of an object that being out of FOV.

5.2.1 Visibility-Based U-Disparity Occupancy Grid

Occupancy grid map is a powerful method for describing an environment and has been used for a variety of applications in robot field. In the last decade, occupancy grids are typically constructed in the Cartesian space from beam-type sensor such as ultrasound or laser range finder. In contrast, stereo camera is conic type sensor, which is less common to build an occupancy grid map due to the needing processing time and the limitation in accuracy [29: Perrollaz et al. 2012]. To overcome these problems, constructing occupancy gird map in u-disparity space using stereo camera is more popular than constructing occupancy gird map in the Cartesian space.

In [29: Perrollaz et al. 2012], u-disparity occupancy grid map is constructed by assuming that each pixel in disparity map is pre-classified as the road or obstacle pixel by double correlation framework proposed in [30: Perrollaz et al. 2010] which exploits different matching hypotheses for vertical and horizontal objects. However, in most applications, occupancy grid map is used to describe the environment without knowing each disparity is obstacle or road. Thus, knowing each pixel is obstacle or free space to construct occupancy does not make sense. Therefore, in this thesis the proposed system modifies the method slightly without pre-categorizing the disparity pixels.

The concept of the visibility-based occupancy grid map considers the ratio between observation pixels and visible pixels in the region of interest (ROI) with the height

according to the disparity of the grid cell. Figure 5.2 shows the concept: assuming that a human stands behind of a car as shown in Figure 5.2(a) with corresponding disparity map, shown in Figure 5.2(b). To estimate the occupancy of the grid cell $U_D = (u, d)$ that the car is located at, the disparity pixels in the region of interest are classified to be visible or non-visible pixels, observed pixels or occluded pixels. u is the image column coordinate, whereas d is the disparity coordinate of the grid at certain distance ($d = fB / Z$). Figure 5.2(c) shows the possible pixels in the region of interest with d as illustrated in Figure 5.2(d), whereas Figure 5.2(e) shows the classification result of these pixels. First, the pixels colored in green are classified as visible with their disparity value smaller than d. This means that these measurement rays pass through the grid cell and do not hit any obstacle (note that the larger disparity d, the smaller distance Z). The pixels colored in yellow are categorized as observed and visible pixels since their value in disparity map are the same as d, which means that the measurements hit the car exactly. The remaining blue pixels are categorized as non-visible pixels. The pixels in occlusion and invalid disparity are all in this case. Figure 5.3(e) better shows the occlusion case: for estimating the grid cell that the human stands at, these pixels do not hit or go through the human, which are occluded by the car in front of the human. These pixels cannot “see” the grid cell and therefore they are classified as non-visible pixels. The relation between these classifications can be

illustrated in Figure 5.2(g).

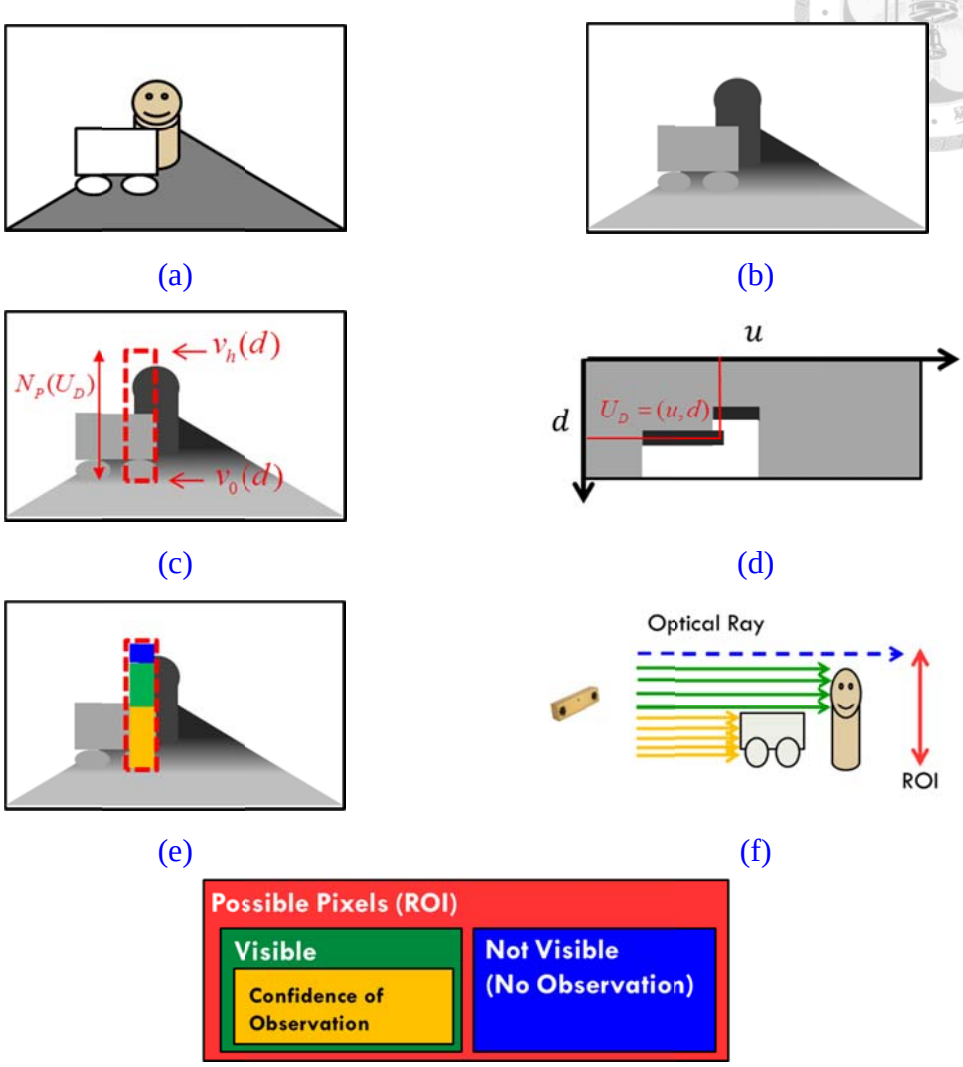


Figure 5.2: Illustration of the visibility-based occupancy grid construction method.

(a)-(b) Color image and corresponding disparity map with a human behind a car.

(c) The range of the possible pixels of the car at certain position $Z = fB/d$, which has the same meaning of region of interest (ROI).

(d) The u-disparity occupancy grid map of (b). U_D is the coordinate of the cell that the car is located at. Note that u-disparity and image has the same width.

(e) Classification of the possible pixels annotated with different color. The annotations in (e)-(g) have the cooresponding colors for better understanding. Blue line stands for the non-visible pixels. Yellow line represents the pixels exactly hit the obstacle at that grid cell. Both yellow and green lines indicate the visible pixels that do not be occluded at that distance and are not invalid.

(f) Each pixel in the ROI can be thought of as a optical ray.

(g) The relation of these classification.

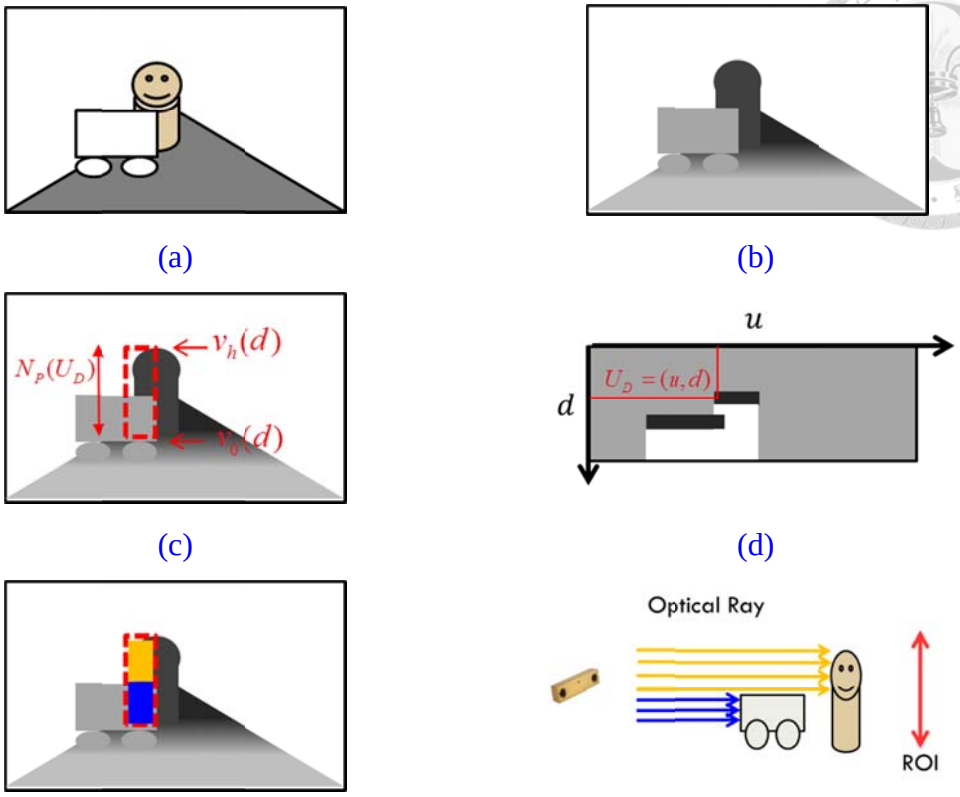


Figure 5.3: Another example to illustrate the concept of the visibility-based occupancy grid map.

(a)-(b) Color image and corresponding disparity map with a human behind a car.

(c) The range of the possible pixels of the human at certain position $Z = fB/d$, which has the same meaning of region of interest (ROI).

(d) The u-disparity occupancy grid map of (b). U_D is the coordinate of the cell that the human is located at.

(e)-(f) Classification of the possible pixels annotated with corresponding color. Blue line stands for the non-visible pixels. Yellow line represents the pixels exactly hit the human at that grid cell. Note that in this example, the hieght of ROI is shortened due to the perspective projection. The blue pixels (blue rays) are categoried as non-visible pixels to the grid cell that the human is located at since these optical rays do not go through the position.

According to [29: Perrollaz et al. 2012], the concepts of the visibility-based occupancy grid construction which are presented above can be formulated as follows [29: Perrollaz et al. 2012]:

$$P(O_U) = \sum_{v,c} P(V_U = v)P(C_U = c)P(O_U \mid V_U = v, C_U = c) \tag{5.1}$$

where $P(O_U)$ is the probability describing the occupancy of a certain grid cell U_D. V_U, C_U and O_U are binary random variables, which can be one of the value in $\{0,1\}$, describing the specific states of the grid U_D. V_U represents the visibility of the cell. $V_U = 1$ means that the grid U_D is visible. C_U indicates the obstacle confidence of the cell. $C_U = 1$ means that an object is seen in the grid U_D. O_U is the occupancy of the cell. $O_U = 1$ shows that the cell is occupied by obstacle pixels.

To solve the Equation (5.1), some boundary conditions of $P(O_U \mid V_U, C_U)$ are intuitive known. First of all, for a grid cell that is in invisible state, the occupancy cannot be determined due to no measurement data. That is, nothing is known about its occupancy. This can be written as follows [29: Perrollaz et al. 2012]:

$$\forall c \in \{0,1\}, P(O_U \mid \neg V_U, C_U = c) = 0.5 \tag{5.2}$$

Secondly, for grid cell that is fully in visible state, the boundary conditions $P(O_U \mid V_U = 1, C_U)$ are determined according to the obstacle confidence state C_U of that grid. If the grid cell is in full confident that an obstacle is observed, this means that the cell is absolutely occupied or is not occupied only when the false positive is

occurred. This can be expressed as follows [29: Perrollaz et al. 2012]:

$$P(O_U \mid V_U, C_U) = 1 - P_{FP} \tag{5.3}$$

On the other hand, if the grid cell was fully visible but noting can be observed, the cell can only be occupied when it occur a false negative. That is:

$$P(O_U \mid V_U, \neg C_U) = P_{FN} \tag{5.4}$$

The four boundary conditions mention above are listed in Table 5.2.

Table 5.2: Bounding conditions of $P(O_U \mid V_U, C_U)$

Visibility / Observed Confident	V_U	$\neg V_U$
C_U	$1 - P_{FP}$	0.5
$\neg C_U$	P_{FN}	0.5

Substituting these boundary conditions in Equation (5.1), it can be extended as follows [29: Perrollaz et al. 2012]:

$$P(O_U) = P(V_U)P(C_U)(1 - P_{FP}) + P(V_U)(1 - P(C_U))P_{FF} + (1 - P(V_U)) \cdot 0.5 \tag{5.5}$$

P_{FN} and P_{FP} can occur during the stereo matching step and are assumed to be a known constant (both of them are 0.02 in this thesis). Therefore, to obtain the occupancy of grid cell $P(O_U)$, the remaining things to be estimated are the visibility of a cell, $P(V_U)$, and the confidence of observation, $P(C_U)$. The visibility is defined as the ratio between the number of visible and possible pixels (length of the ROI), that is [29: Perrollaz et al. 2012]:

$$P(V_U) = \frac{N_V(U_D)}{N_P(U_D)} \tag{5.6}$$

where $U_D = (u, d)$ stands for the certain grid cell in u-disparity space. $N_P(U_D)$ is the number of possible pixels at the cell depends on its d-coordinate, which is defined as follows [29: Perrollaz et al. 2012]:

$$N_P(U_D) = v_h(d) - v_0(d), \tag{5.7}$$

where $v_h(d)$ is the v-coordinate of the pixel which are situated at the maximum detection height for certain disparity d. Similarly, $v_0(d)$ is the v-coordinate of the pixel which are located on the ground for certain disparity d. $v_0(d)$ and $v_h(d)$ can be obtained by the fundamental pin-hole model and are expressed as follows:

$$v_h(d) = d\frac{Y_h}{B} + v_{center} \tag{5.8}$$

$$v_0(d) = d\frac{Y_{ground}}{B} + v_{center} \tag{5.9}$$

where B is the baseline of the stereo camera and v_{center} is the v-coordinate of the center of the image plane. Y_h and Y_{ground} are. Substituting Equation (5.8) and (5.9) into (5.7), it becomes:

$$\begin{aligned} N_P(U_D) &= d\frac{Y_h}{B} + v_{center} - d\frac{Y_{ground}}{B} + v_{center} \\ &= d\frac{(Y_h - Y_{ground})}{B} \\ &= dC_{ROI} \end{aligned} \tag{5.10}$$

C_{ROI} is a constant which depends on the preset detection height and ground position. Thus, $N_P(U_D)$ only depends on the d-coordinate for a given cell U_D.

On the other hand, $N_V(U_D)$ is the number of visible pixels in the subset of the

possible pixels, which can be expressed as:

$$N_V(U_D) = \sum_{v=v_0}^{v_h} (F_{d,visible}(u,v)) \,, \tag{5.11}$$

$$F_{d,visible} = \begin{cases} 1, if\ \ I_d(u,v) <= d\ \&\&I_d(u,v) \neq 'invalid' \\ 0, other \end{cases} \tag{5.12}$$

With these expressions, the visibility of a grid cell $P(V_U)$ can be obtained as Equation (5.6). For estimating the confidence of the observation, $P(C_U)$, the ratio between the observed pixels and the visible pixels is considered. It is defined as an exponential function as follows [29: Perrollaz et al. 2012]:

$$P(C_U) = 1 - e^{-\frac{r_O(U_D)}{\tau_O}} \tag{5.13}$$

where τ_O is a constant and is chosen to be $\tau_O = 0.1$ as the same value in [29: Perrollaz et al. 2012], and r_O is defined as the obstacle confidence, which is the ratio between observed pixels and the visible pixels, that is, [29: Perrollaz et al. 2012]:

$$r_O(U_D) = \frac{N_O(U_D)}{N_V(U_D)} \tag{5.14}$$

The number of the observed pixels can be expressed as follows:

$$N_O(U_D) = \sum_{v=v_0}^{v_h} (F_{d,observed}(u,v)) \tag{5.15}$$

$$F_{d,observed} = \begin{cases} 1, if\ \ \text{abs}(I_d(u,v) - d) <= C_{ObservedThreshold}\ \&\&I_d(u,v) \neq 'invalid' \\ 0, other \end{cases} \tag{5.16}$$

For all the grid cells $U_D = (u,d)$ in the u-disparity space, their occupancy can be obtained by the above expressions, Equation (5.5), (5.6) and (5.13). Note that $N_P(U_D)$, $v_h(d)$ and $v_0(d)$ of each cell can be pre-calculated since they will not change during

the occupancy grid calculation processing. Hence this pre-calculation process reduces the time complexity and then speeds up the overall algorithm.

5.2.2 Post-processing

The raw u-disparity occupancy grid is constructed by using visibility-based calculation method, as mentioned in Subsection 5.2.1. The occupancy grid can be marked as 'occupied' by using a probability threshold to extract object candidate. However, due to the noisy data from stereo camera and the discretization effects, applying simple threshold without any preprocess will cause many unwanted candidate results. Thus, before applying threshold to extract "occupied" grid cell, a series of image processes are applied to refine the raw occupancy grid. The overall post-processing flow chart is shown in Figure 5.4.

First of all, in order to eliminate spatial noise to obtain smoother and more realistic representation, 2-D Gaussian filtering is applied. Since the standard deviation σ_d along the d axis is related to the disparity discretization, for example $\sigma_d^2 = 0.5$, and the standard deviation σ_u along the u axis is related to the width of the correlation window to model effects like foreground fattening in the u-disparity plane [30: Perrollaz et al. 2010], the 5×5 2-D Gaussian filter mask is selected, as shown in Figure 5.4(d), which is a constant Gaussian kernel. This 5×5 mask acts as image filter based on convolution which can be written as Equation (5.8). Therefore this process is much fast

and easy to implement.

$$P^*(O_U) = G(\sigma_u, \sigma_d) * P(O_U) \tag{5.17}$$

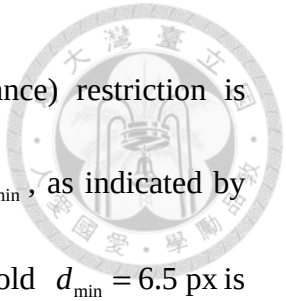
After removing spatial noise by using Gaussian filter, the cells can be flagged as "occupied" by applying a constant probability threshold. That is,

$$B(i, j) = \begin{cases} 1, & if \ P^*(O_U(i, j)) \geq Threshold \\ 0, & \text{other} \end{cases} \tag{5.18}$$

By doing so, the result of the u-disparity occupancy grid is a binary image, as shown in Figure 5.4(f). This occupied mask might have a lot of disconnect part that cannot be filtered out by using Gaussian filtering. This phenomenon is illustrated in Figure 5.4(e)-(f). To overcome this problem, a morphological image process is applied to reduce the remaining noise and link the disconnect part, as shown in Figure 5.4(g). Some of the small noise will be reduced, while the larger near neighbors are linked together and formed as a connected component to solve disconnection problem.

The next step is to extract object candidates by using the connected-component labeling with 4-connectivity as mentioned in Section 3.3.4. Figure 5.4(h) shows the connected-component labeling result of the obstacle grid map, Figure 5.4(g), each region is marked in different colors and indicates an object candidate C_j.

For farther distance object, the number of pixels projected from 3D coordinate to image plane is too sparse to analyze the image information of object candidate due to perspective projection. Moreover, the large uncertainty of stereo camera in far distance



will cause inaccurate result. Therefore, a simple disparity (distance) restriction is applied to remove the candidate whose disparity is smaller than $d_{\min}$, as indicated by the red line shown in Figure 5.4(h). The minimum disparity threshold $d_{\min} = 6.5$ px is used in this thesis. In addition, object candidate is assumed to have image width larger than a certain threshold and is filtered by the following expression:

$$W(C_j) < W_{Th} \tag{5.19}$$

where $W_{Th} = 25\ pixels$ is used in this thesis.

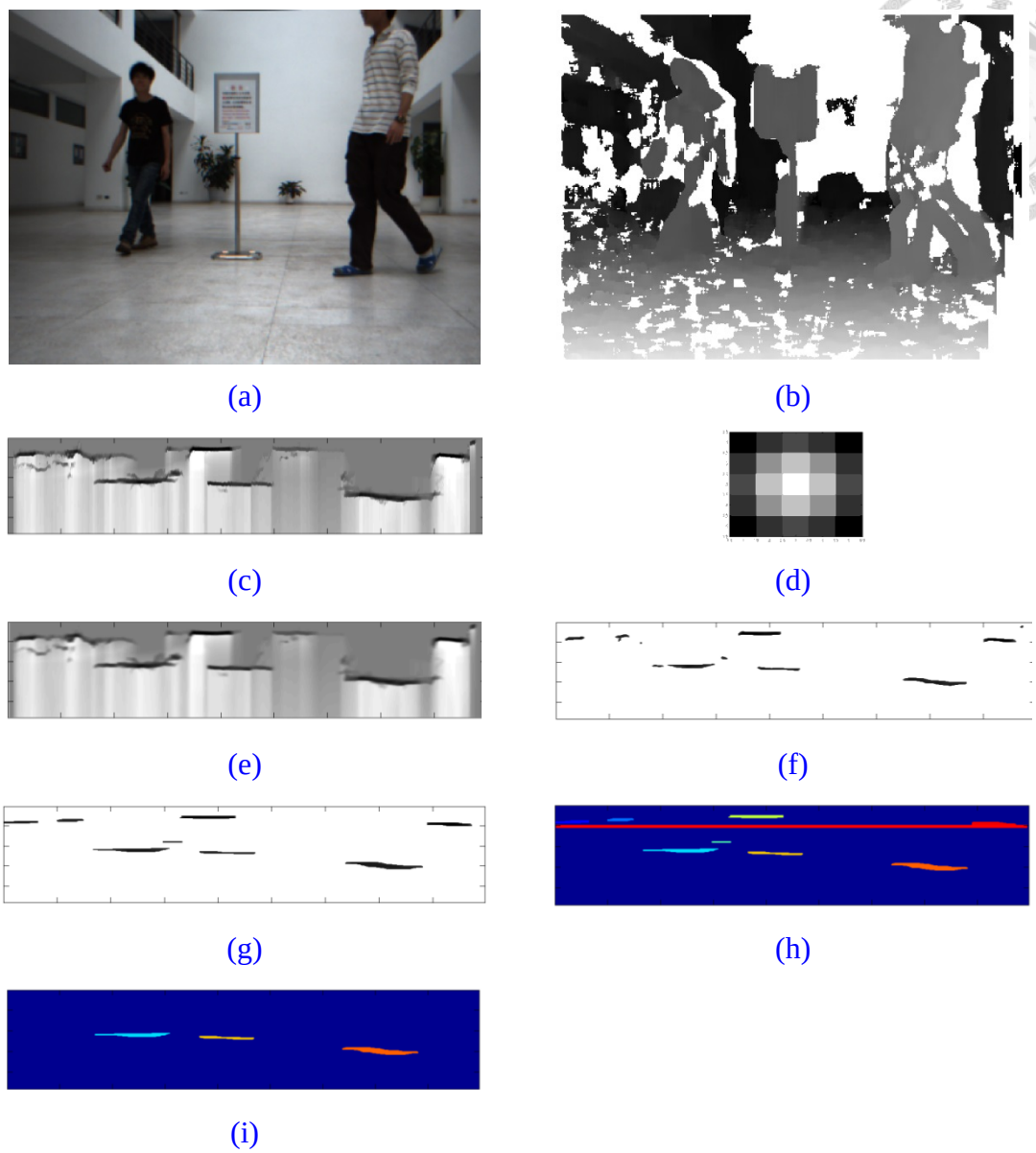


Figure 5.4: Each post-processing step applies to the u-disparity occupancy grid.

(a) Image from stereo target camera.

(b) Corresponding disparity map.

(c) The original u-disparity occupancy grid. Note that the darker pixel means the higher probability of being occupied.

(d) An example of the constant 2-D Gaussian kernel filter with size 5×5.

(e) The occupancy grid filtered by 2-D Gaussian kernel.

(f) Occupied binary mask by applying a certain probability threshold to (e)

(g) Modified occupied mask by applying a series of morphological processes.

(h) Connected-component labeling of (h). The red line indicates the disparity threshold $d_{\min}$.

(i) Remove the candidate whose disparity $d > d_{\min}$ and short candidate in (h)

5.2.3 Object Candidates Bounding Box Extraction

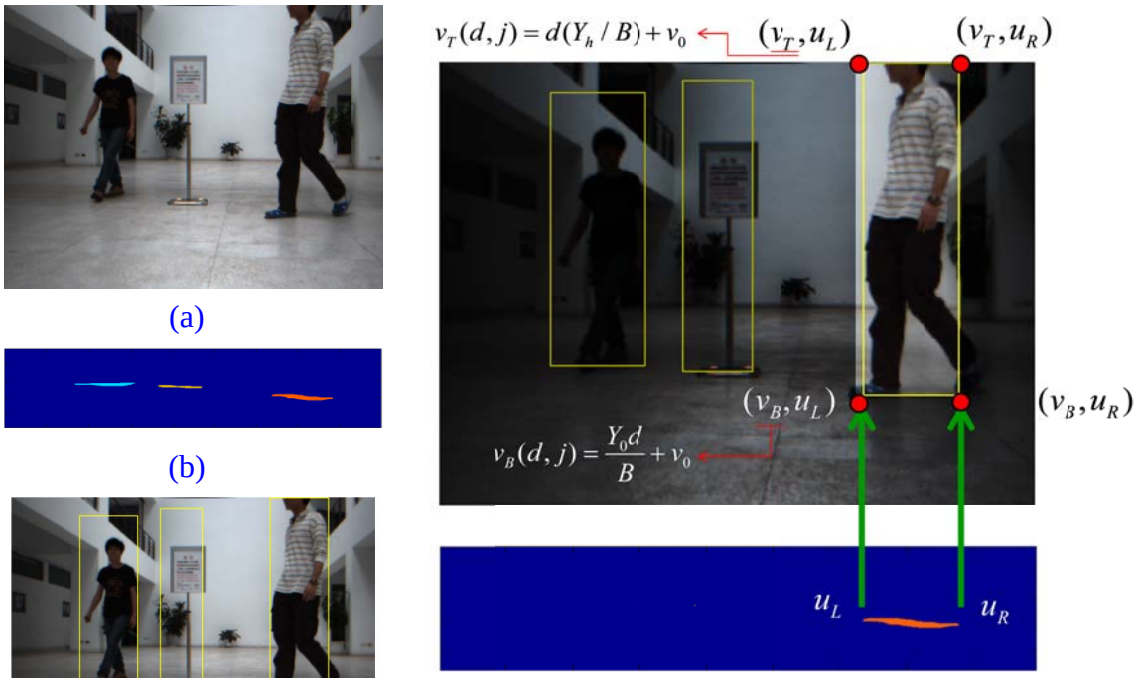
As mentioned in section 5.2, the u-disparity occupancy grid is constructed by using visibility-based occupancy grid method. Higher value of a grid cell means that the grid has large probability that an object is located. Therefore, the 'obstacle' grid can be extracted by using simple probability threshold. The obstacle grid is a binary image that indicates each grid cell is occupied or not. Using connected-component labeling on the obstacle grid can extract object candidate in u-disparity space. To describe the candidate in the image coordinate, a rectangle called bounding box is used, which need four variables, v_T, v_B, u_L and u_R, to indicate the position of the vertexes, as shown in Figure 5.5(c). Since the u-disparity and image width has same coordinate, which is the image column, the u-coordinate of the bounding box can be extracted from the pixels in u-disparity grid directly. The v-coordinates, however, cannot be obtained directly from u-disparity and is calculated by the d-coordinate d with the pre-defined positions of the ground Y_0 and maximum detection height Y_h as the same in the Equation (5.8) and (5.9) in the Section 5.2.1. The v-coordinate of the top two vertexes is calculated by Equation (5.20), whereas the v-coordinate of the bottom two is calculated by Equation (5.21).

$$v_T(d, j) = \frac{Y_h f}{Z} + v_0 = \frac{Y_h d}{B} + v_0 \qquad (5.20)$$

$$v_B(d,j) = \frac{Y_0 f}{Z} + v_0 = \frac{Y_0 d}{B} + v_0 \quad (5.21)$$

$$Z = \frac{fB}{d} \quad (5.22)$$

By doing so, each candidate labeled in u-disparity space can find the corresponding bounding box to reach the information from image and disparity map. The bounding box extraction result is shown in Figure 5.5(d).



(a)

(b)

(c)

(d)

Figure 5.5: Illustration of bounding box extratction from u-disparity obstacle grid.

5.3 Object Tracking

Object candidates are extracted in previous section, the next problem is how to link database object to corresponding candidate measurement in the current step correctly. It is so called “data association” problem in the field of robotic. For sensors that only provide range information, it is difficult to do data association because no other information can use to distinguish each of measurement points. Using sensor that provides RGB-D data such as stereo camera can easily handle this problem since these sensors provide additional image information to describe an object. This chapter will describe the proposed solution to solve the data association problem in detail.

5.3.1 Remove Background Pixels in Bounding Box

Since many pixels do not belong to the object in the bounding box of detected candidates, analyzing all the pixels of the bounding box directly without filtering out the non-object pixels would get the wrong result. Therefore, before analyzing the object pixels that within in the bounding box, background pixels should be removed. The background pixels can be seen as the pixel with depth value that do not in the range of the uncertainty in that distance. That is:

$$B(i,j)=\begin{cases}1, & \overline{Z}_{object}-\Delta Z<Z(i,j)<\overline{Z}_{object}+\Delta Z\\ 0, & other\end{cases} \tag{5.23}$$

$B(i,j)$ is a binary image which indicates if the position (i,j) is foreground or

background pixel. $\overline{Z}_{object}$ is the depth value retrieved from u-disparity occupancy grid, and the uncertainty ΔZ can be formulated as follows [49: Accuracy For Stereo Vision from PointGrey 2010]:

$$\Delta Z = \left| \frac{fB}{d} - \frac{fB}{d+\Delta d} \right| = \frac{fB}{d}\frac{\Delta d}{(d+\Delta d)} = \overline{Z}_{object}\frac{\Delta d}{(\frac{fB}{\overline{Z}_{object}}+\Delta d)} = \frac{\overline{Z}^2_{object}\Delta d}{(fB+\overline{Z}_{object}\Delta d)} \quad (5.24)$$

where $\Delta d = matching\ error = 0.1$.

Moreover, the object is assumed to have thickness $Z_{Thickness}$, therefore Equation (5.11) can be rewritten as follows:

$$B(i,j) = \begin{cases} 1, & \overline{Z}_{object} - \Delta Z - Z_{Thickness} < Z(i,j) < \overline{Z}_{object} + \Delta Z + Z_{Thickness} \\ 0, & other \end{cases} \quad (5.25)$$

where the thickness is assumed to be $Z_{Thickness} = 0.3m$.

The position (i, j) in image plane will be flagged as object pixel (foreground) when the corresponding depth value $Z(i, j)$ is in the uncertainty range, as illustrated in Figure 5.6. These remaining pixels are used to be the input of the HSV histogram.

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

Figure 5.6: Background Pixels Removal for frame #236.
(a) Bounding box image
(b) Bounding box depth map
(c) Depth map with background removal
(d) Foreground mask
(e) Image with mask filtering

5.3.2 Registration between Candidates and Objects using Feature Vectors

To distinguish one object to another, distinctive and invariant information that can describe an object should be selected. Therefore, the images of a certain object in different frames are analyzed at first. Figure 5.8 and Figure 5.9 show the related information about the object such as the histograms of R, G, B, H, S and V channels in frame No. 236 and No. 191, respectively for example. It can be observed that the distributions of the H and S histograms are time-invariant, while the other channels are changed dramatically. Moreover, assuming that object moves consequently, the histogram distribution of H and S will not change dramatically, as shown in Figure 5.8 and Figure 5.11. On the other hand, the other object and the corresponding information in frame No. 236 are shown in Figure 5.10 and are compared to Figure 5.8. It can be observed that different object has corresponding distribution on H and S histogram. Since the distributions of H and S histograms have distinctive and invariant properties, data association problem can be solved by using H and S histograms as feature vectors to register the certain object in different frame images. In this thesis, each H and S channel is binned into 64 intervals to form a 1×64 feature vector. Note that H and S histograms are normalized to form a probability distribution representation, that is,

$$P_H(O_i) = \frac{\mathbf{H}_H(O_i)}{N} \tag{5.26}$$

$$P_S(O_i) = \frac{\mathbf{H}_S(O_i)}{N} \tag{5.27}$$

where N is the remaining pixel number in the bounding box, O_i indicate the i-th object, $\mathbf{H}_H(O_i)$ represents the histogram of the H channel of the i-th object, $\mathbf{H}_S(O_i)$ is the histogram of the S channel of O_i.

To achieve the goal of registering candidate C_j to corresponding database object O_i successfully, estimating the similarity between object and candidate feature vectors using Bhattacharyya distance is applied. The Bhattacharyya distance is a similarity index that measures two probability distributions. The definition of Bhattacharyya distance is as follows [41: Comaniciu et al. 2003]:

$$d(p,q) = \sqrt{1 - BC(p,q)} \tag{5.28}$$

where BC is the Bhattacharyya coefficient, which is defined as follows:

$$BC(p,q) = \sum_{i=1}^{N} \sqrt{p(i)q(i)} \tag{5.29}$$

To calculate the Bhattacharyya distance between the H channel distribution of candidate $P_H(C_j)$ and the distribution of database object $P_H(O_i)$, $P_H(C_j)$ and $P_H(O_i)$ are substituted into Equation (5.28) and (5.29), the equations become:

$$d_H\left(P_H(C_j), P_H(O_i)\right) = \sqrt{1 - BC(P_H(C_j), P_H(O_i))} \tag{5.30}$$

$$BC\left(P_S(C_j), P_S(O_i)\right) = \sum_{m=1}^{M=64} \sqrt{P_{S,m}(C_j) P_{S,m}(O_i)} \tag{5.31}$$

Similarly, for the Bhattacharyya distance between the S channel of candidate and object,

the equations become:

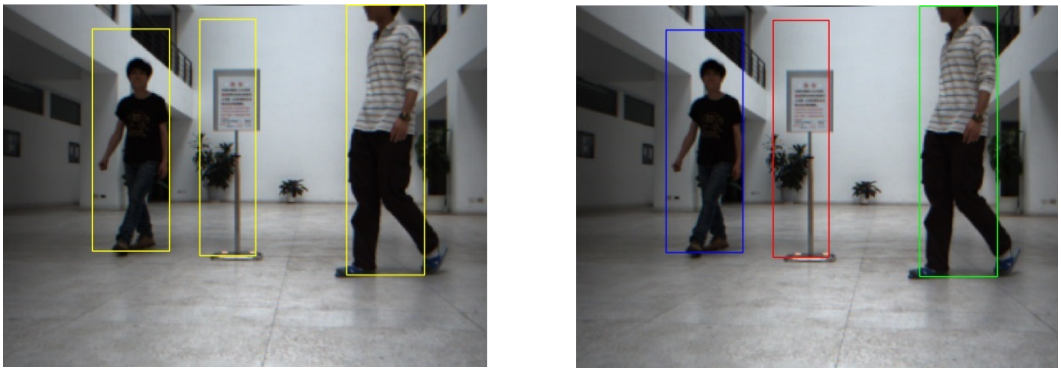
$$d_S\left(P_S(C_j),P_S(O_i)\right)=\sqrt{1-BC(P_S(C_j),P_S(O_i))} \tag{5.32}$$

$$BC\left(P_H(C_j),P_H(O_i)\right)=\sum_{m=1}^{M=64}\sqrt{P_{H,m}(C_j)P_{H,m}(O_i)} \tag{5.33}$$

Combining these two similarity index with same weights, the total Bhattacharyya distance between j-th candidate and i-th objects is expressed as follows:

$$d_{Total}\left(C_j,O_i\right)=d_H\left(P_H(C_j),P_H(O_i)\right)+d_S\left(P_S(C_j),P_S(O_i)\right) \tag{5.34}$$

Therefore, each candidate can be registered to the database object successfully, as shown in Figure 5.7.



(a) (b)

Figure 5.7: The registration result of each candidate to the database object. Different objects are enclosed by different color bounding boxes

(a) Candidates before registration.

(b) Candidates after registration.

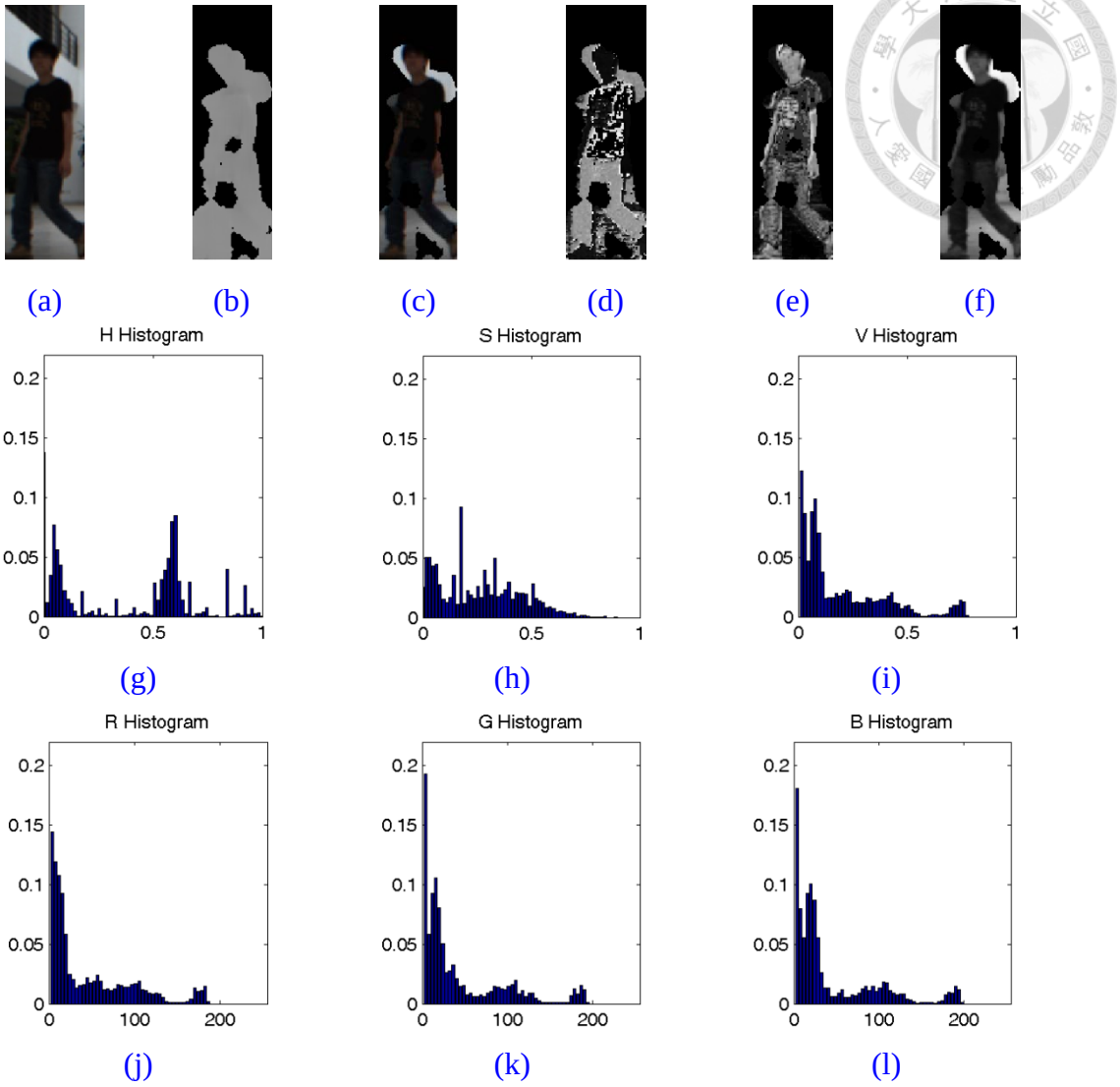


Figure 5.8: Properties of object 1 in frame 236

(a) Bounding box image.

(b) Depth image with background removal.

(c) Image with background removal.

(d)-(f) H, S and V channel of HSV image transformed from (c).

(g)-(i) H,S and V channel histogram

(j)-(l) R,G and B channel histogram

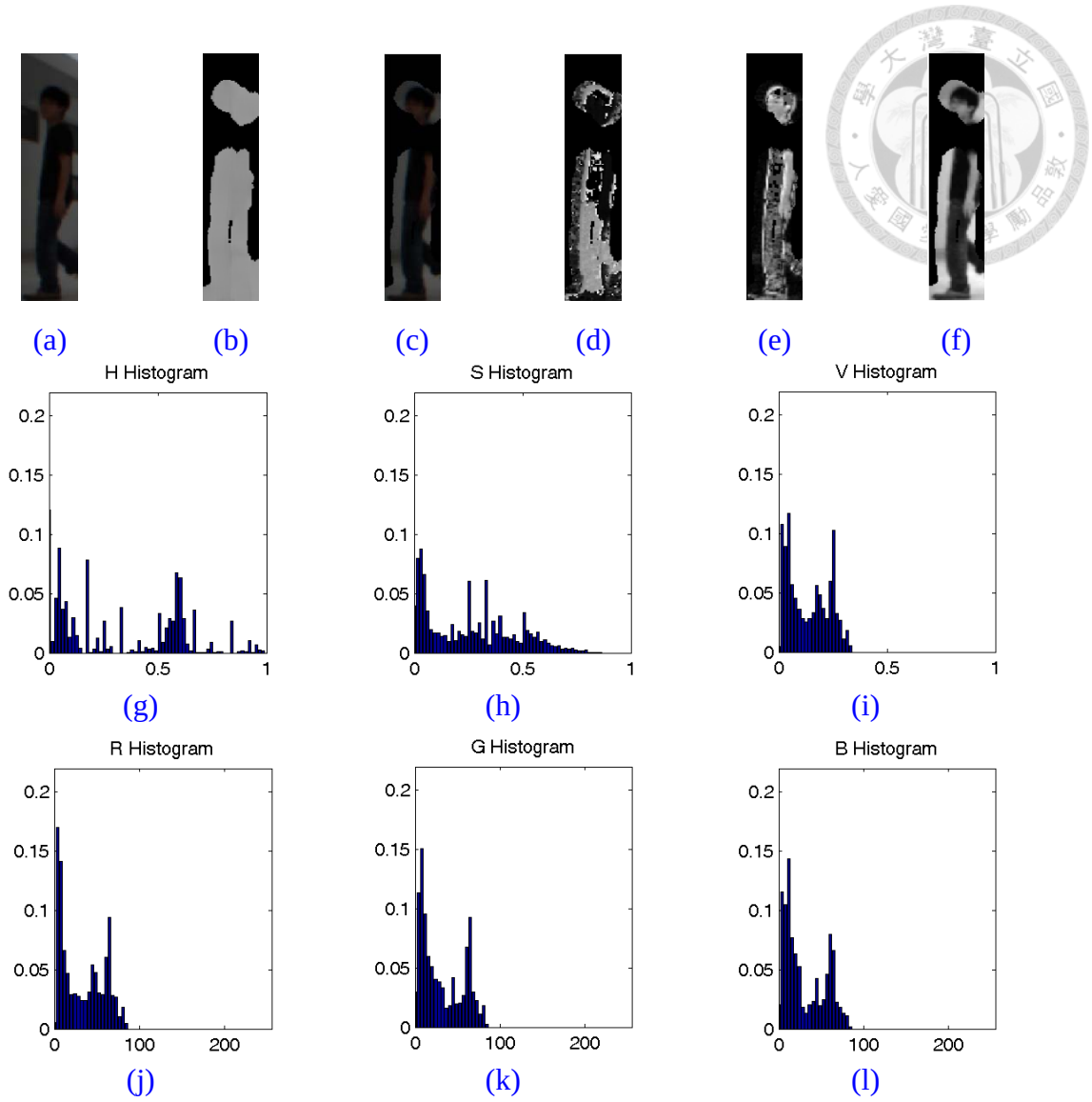


Figure 5.9: Properties of object 1 in frame 191

(a) Bounding box image.

(b) Depth image with background removal.

(c) Image with background removal.

(d)-(f) H, S and V channel of HSV image transformed from (c).

(g)-(i) H,S and V channel histogram

(j)-(l) R,G and B channel histogram

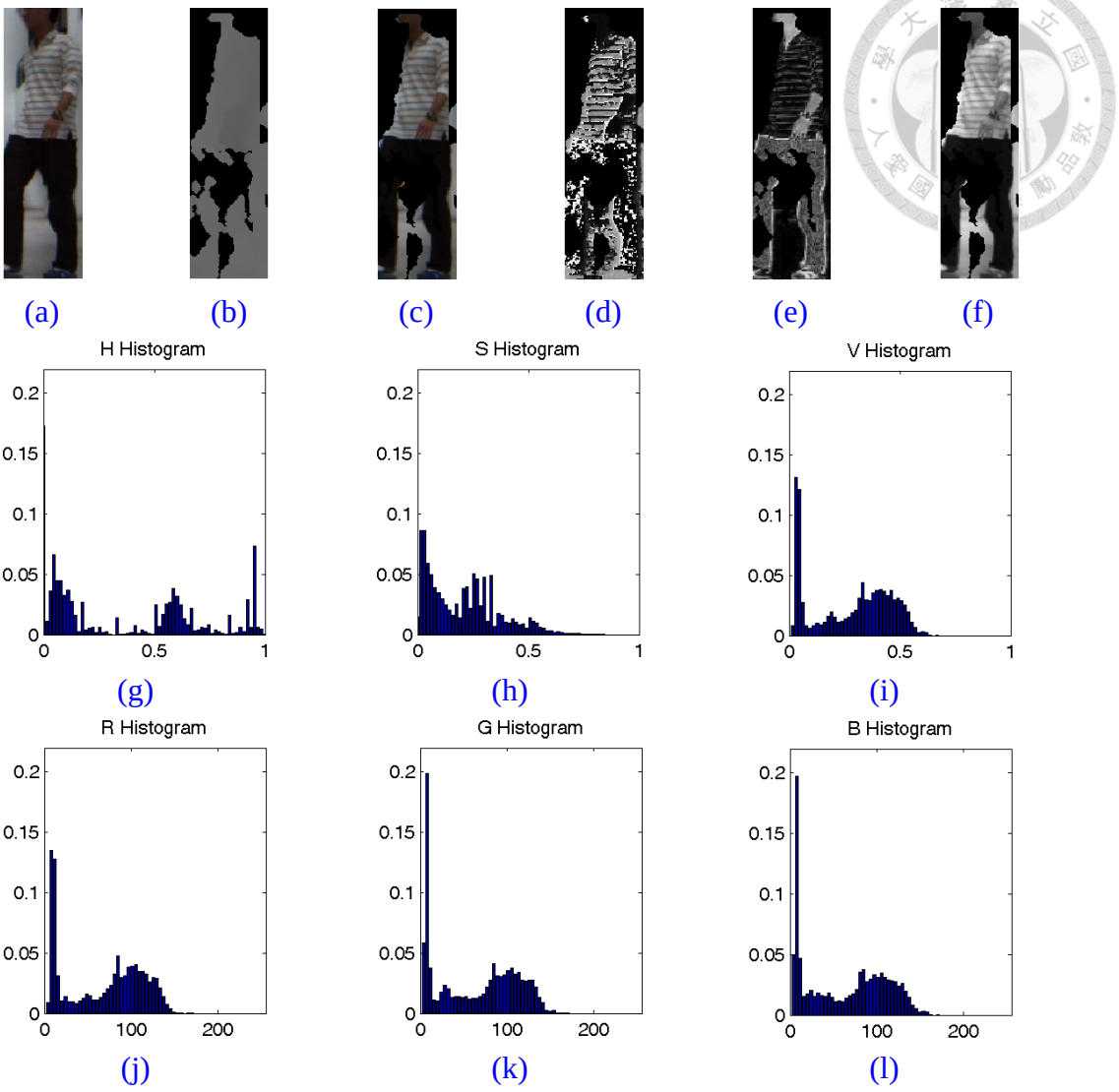


Figure 5.10: HSV histogram of object 2 in frame 236

(a) Bounding box image.

(b) Depth image with background removal.

(c) Image with background removal.

(d)-(f) H, S and V channel of HSV image transformed from (c).

(g)-(i) H,S and V channel histogram

(j)-(l) R,G and B channel histogram

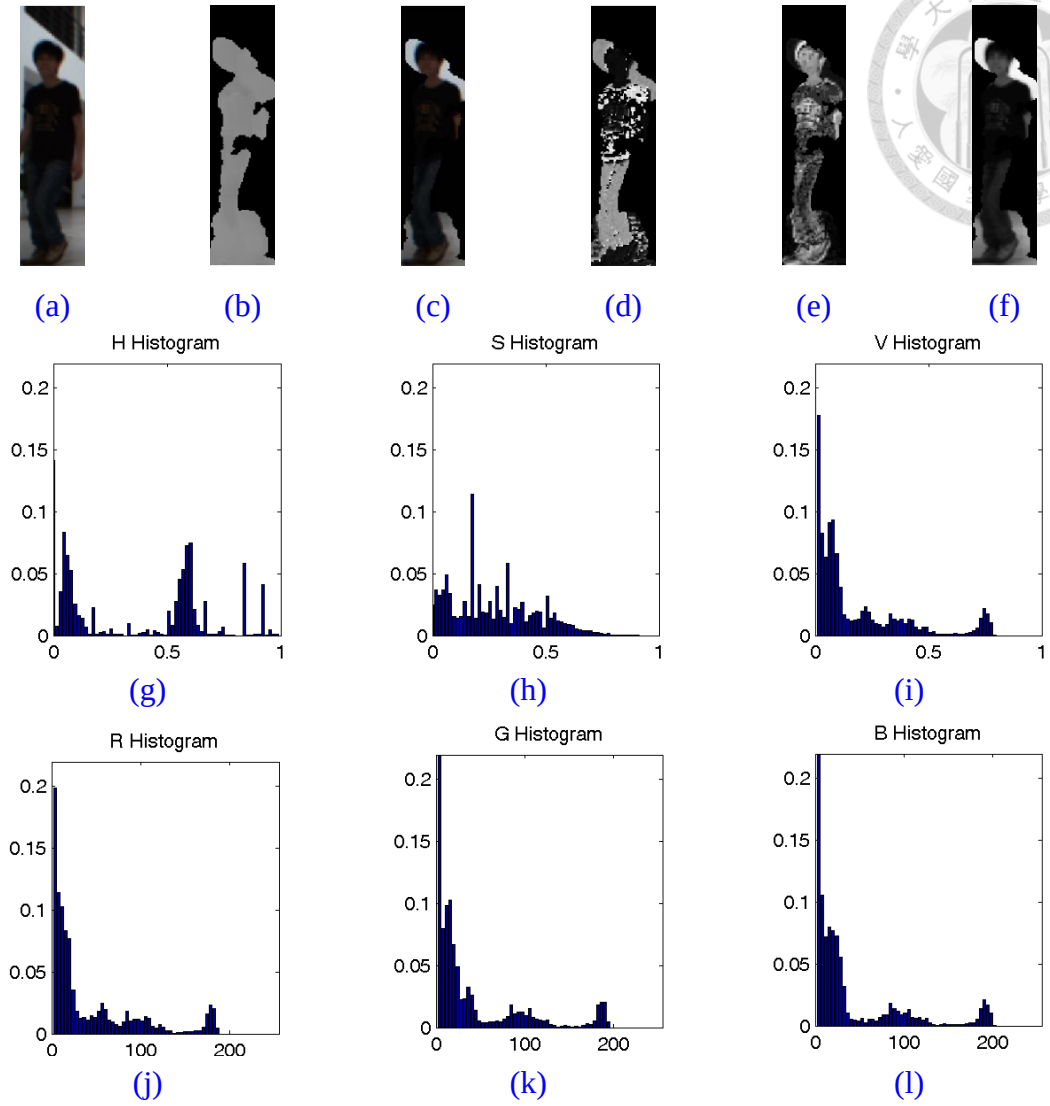


Figure 5.11: Properties of object 1 in frame 237

(a) Bounding box image.

(b) Depth image with background removal.

(c) Image with background removal.

(d)-(f) H, S and V channel of HSV image transformed from (c).

(g)-(i) H,S and V channel histogram

(j)-(l) R,G and B channel histogram

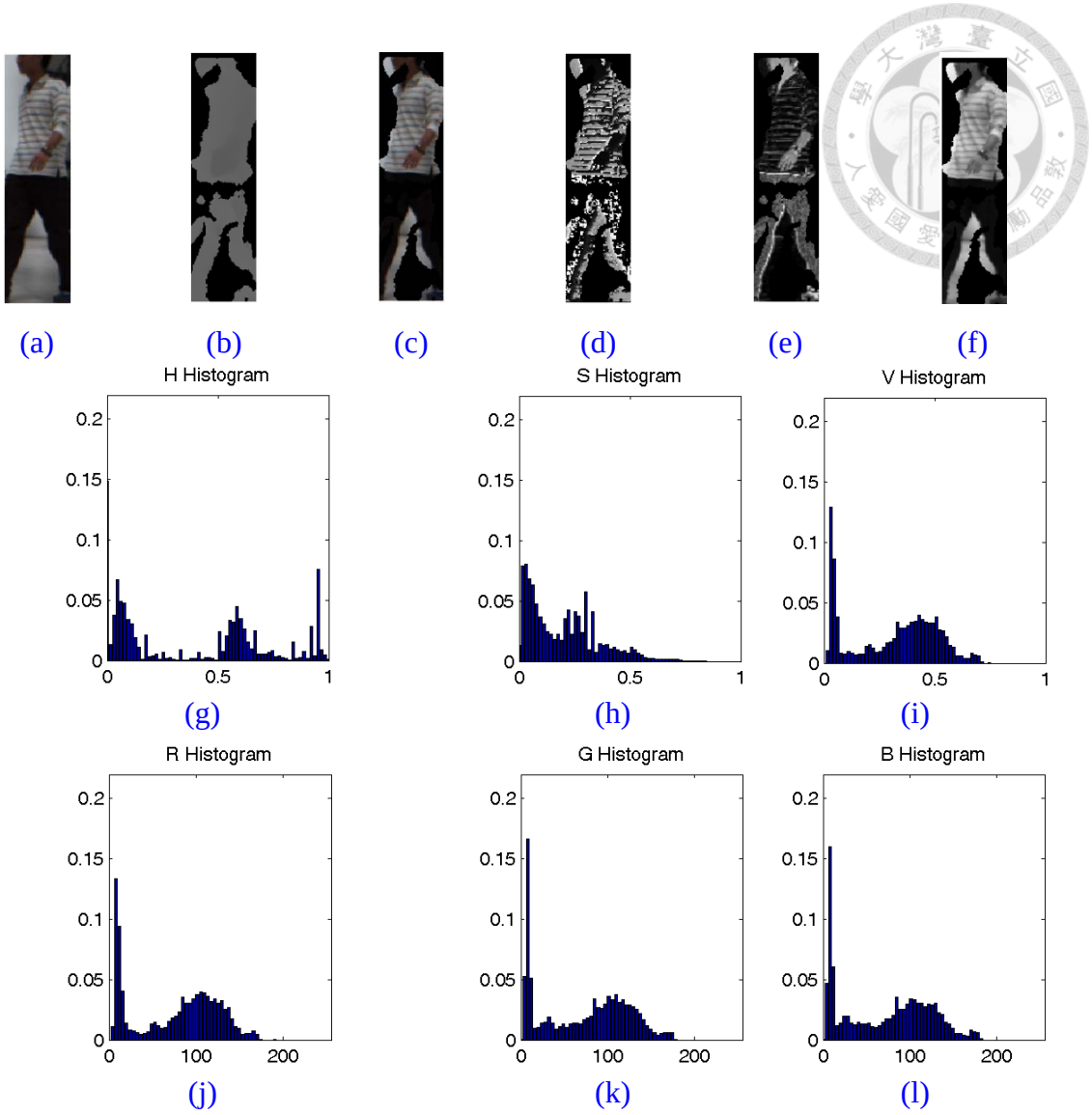


Figure 5.12: Properties of object 2 in frame 237

(a) Bounding box image.

(b) Depth image with background removal.

(c) Image with background removal.

(d)-(f) H, S and V channel of HSV image transformed from (c).

(g)-(i) H,S and V channel histogram

(j)-(l) R,G and B channel histogram

Spatial Constraint

The data association result using hue and saturation histograms as feature vectors is suitable in this thesis. However, considering the situation which two objects are too similar to cause wrong registration result, spatial constraint is added. Since object will not move dramatically, the database object is restricted in a region. Here a radius threshold is used. If all database objects position were not in the range, the candidate is new coming object in database. To check database objects is in the circular range or not, the Euclidean distance is used. That is, if the distance from i-th object to candidate was smaller than r, the candidate can be register to the i-th object.

$$\sqrt{(x_{i,k} - m_{x,j,k}) + (z_{i,k} - m_{z,j,k})} < r \tag{5.35}$$

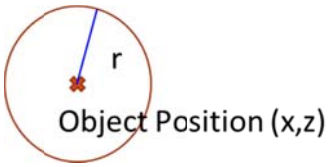


Figure 5.13: A radius distance threshold for the possible range of the object candidates.

5.3.3 Update Strategy with Kalman Filter

After the database objects are linked to the current candidates correctly, the object states can be updated by the measurements. However, database object may not register successfully due to out of field of view or occlusion, the update strategy is proposed to handle the update problem. Figure 5.15(a) shows the case of an object moving out of the field of view of the camera, whereas Figure 5.15(b) shows the situation that an object is occluded by another. Moreover, Kalman filter with constant velocity is used to estimate the object state. The proposed update strategy is shown in Figure 5.14.

First of all, the updater inspects if the object is measured by stereo camera or not by checking the state of the object measurement flag, $F_{i,Measurement}$. When a database object is successfully registered to a candidate as mentioned in Section 5.2.2, the object measurement flag $F_{i,Measurement}$ is set to 1. If the database object has measurement, the next question is if the object is in normal, occlusion or in the camera FOV in previous state. If the object is occluded or out of the camera FOV, it does not have previous position and velocity states, and thus cannot predict its current position and can only rely on the measurement information, this can be expressed as follows:

$$\boldsymbol{x}_{i,k} = \boldsymbol{m}_j \rightarrow \begin{pmatrix} x_{i,k} \\ z_{i,k} \end{pmatrix} = \begin{pmatrix} m_{x,j} \\ m_{z,j} \end{pmatrix} \tag{5.36}$$

$$\boldsymbol{v}_{i,k} = (\boldsymbol{x}_{i,k} - \boldsymbol{x}_{i,k-1})\Delta t^{-1} \rightarrow \begin{pmatrix} v_{i,k,x} \\ v_{i,k,z} \end{pmatrix} = \begin{pmatrix} (x_{i,k} - x_{i,k-1}) / \Delta t \\ (z_{i,k} - z_{i,k-1}) / \Delta t \end{pmatrix} \tag{5.37}$$

If the object is not occluded and is in the camera FOV, Kalman filter can apply to this

case to track the object state. This process is divided into two steps, motion prediction and measurement update. In this thesis, the constant velocity model is applied to describe an object motion. The object motion state and corresponding covariance prediction can be expressed as follows:

$$\overline{\boldsymbol{X}}_{i,k} = F_k \boldsymbol{X}_{i,k-1} \rightarrow \begin{bmatrix} \overline{x}_{i,k} \\ \overline{z}_{i,k} \\ \overline{v}_{i,k,x} \\ \overline{v}_{i,k,z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i,k-1} \\ z_{i,k-1} \\ v_{i,k-1,x} \\ v_{i,k-1,z} \end{bmatrix} \tag{5.38}$$

$$\overline{P}_k = F_k P_k F_k^T + R \tag{5.39}$$

$\overline{\boldsymbol{X}}_{i,k}$ is the i-th object state prediction. F_k is a constant velocity state transition matrix, which is extended as Equation (5.38). $\boldsymbol{X}_{i,k-1}$ is the object state at previous time step. $\overline{P}_k$ is the covariance prediction, whereas R is the motion noise covariance.

For the measurement update step, it combines the state prediction result and the measurement to the object. This process can be expressed as follows:

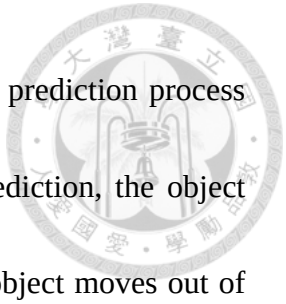
$$K_k = \overline{P}_k H^T (H \overline{P}_k H^T + Q)^{-1} \tag{5.40}$$

$$\boldsymbol{X}_{i,k} = \overline{\boldsymbol{X}}_{i,k} + K_k (Z_{i,k} - H \overline{\boldsymbol{X}}_{i,k}) \tag{5.41}$$

$$P_k = (I - K_k H) \overline{P}_k \tag{5.42}$$

K_k is the Kalman gain. H is the measurement matrix with size 2×4. Q is the covariance of the measurement noise. $\boldsymbol{X}_{i,k}$ is the final update state of the i-th object, whereas P_k is the object state covariance update result.

On the other hand, if the object has no measurement in current step, only when the



object is in camera FOV and is not occluded can apply the motion prediction process which is similar to Equation (5.38). After finishing the motion prediction, the object may move out of the camera FOV or be occluded. To check if the object moves out of the FOV, the angle of object-to-camera is calculated by using the inverse tangent of x / z as illustrated in Figure 5.16, that is,

$$\theta_{ObjToCam} = \arctan(\frac{x}{z}) \tag{5.43}$$

If $\theta_{ObjToCam}$ is larger than the half of the camera FOV, then the object is not in the field of view of the camera and $F_{i,InImage}$ is set to 0. That is,

$$abs(\theta_{ObjToCam}) > \left(\frac{1}{2} \times CameraFOV\right) = 21.5^{\circ} \tag{5.44}$$

A counter $Cnt_{i,FOV}$ is used to check the continuity in order to solve the case that object moves at the boundary of the camera FOV. Here the counter threshold is set to 3.

If $\theta_{ObjToCam}$ is smaller than the half of the camera FOV, it is considered to be occluded, and the occlusion flag $F_{i,Occlude}$ is then set to 1.

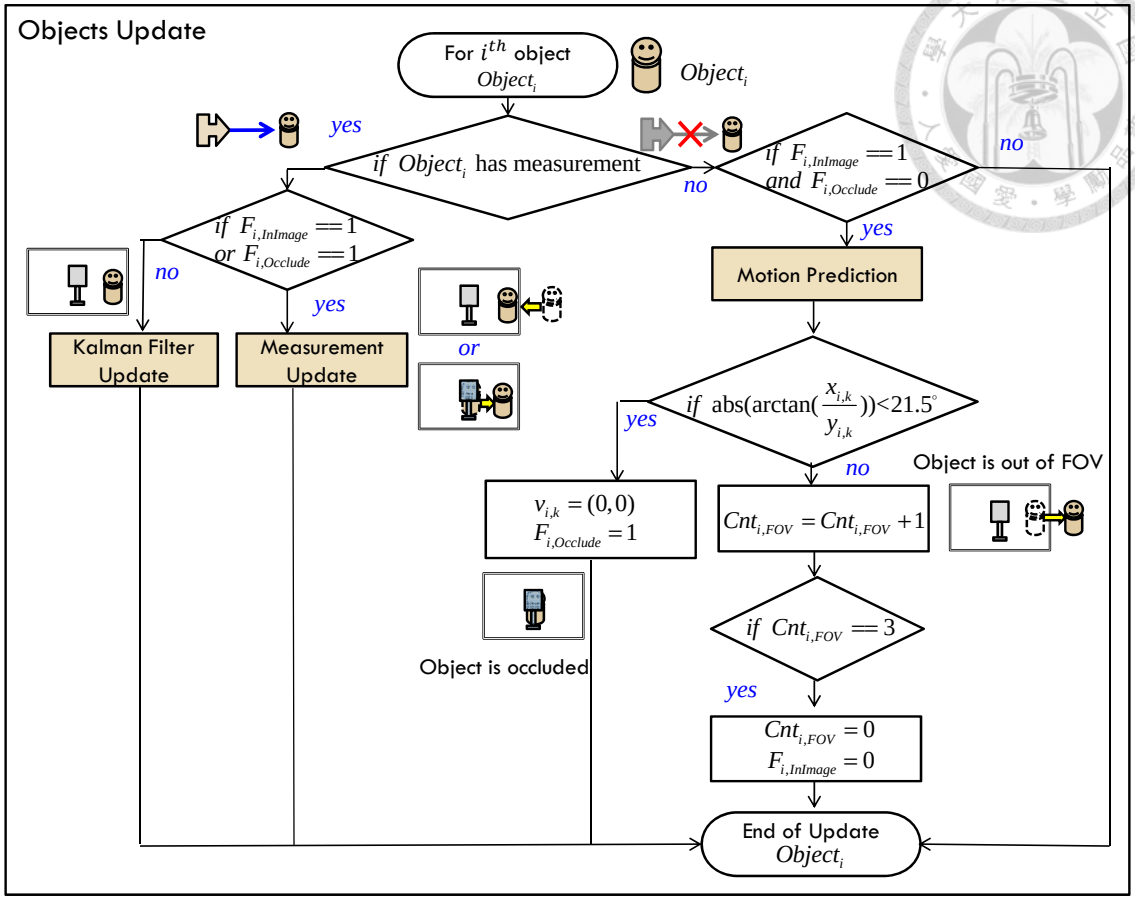
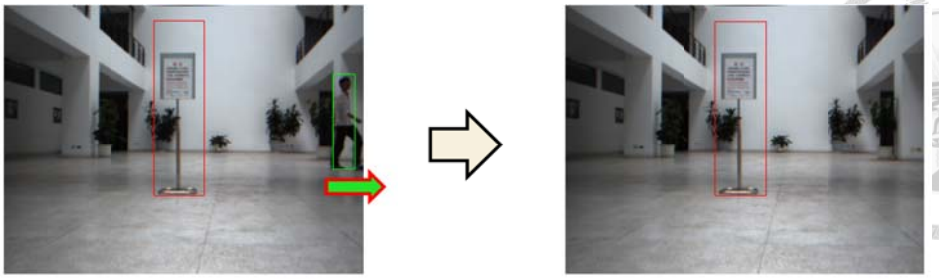


Figure 5.14: The proposed object update flowchart.



(a)

(b)

Figure 5.15: Two cases of unsuccessful object registration since no measurement in current step.

(a) An object moves out of the field of view of the camera.

(b) An object is occluded by another object.

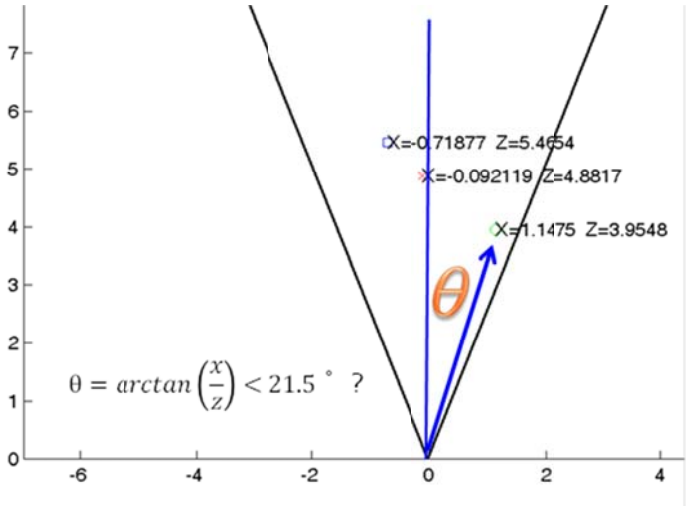


Figure 5.16: Checking if the object is out of the camera field of view (FOV).

Chapter 6
Experiment Result and Analysis

The proposed methods which are presented in Chapter 4 and Chapter 5 are tested in this chapter. In the beginning of this chapter, the apparatuses are shown in the Section 6.1. Experimental results and analysis of three-dimensional localization and mapping algorithms which is proposed in this thesis are discussed in Section 6.2. The result and analysis of the proposed stereo refinement algorithm presented in Section 4.2 are shown in Section 6.3. For the proposed object detection and tracking system, the experimental result and analysis are shown in Section 6.4.

6.1 Experimental Hardware

Stereo Camera

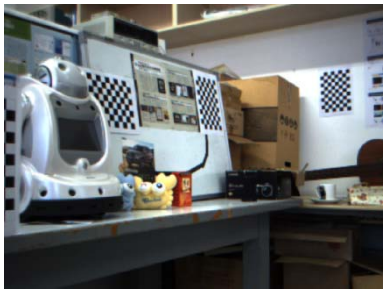
The stereo camera used in this thesis is Point Grey Bumblebee2 BB2-03S2-60, which is shown in Figure 6.1. According to the online specification sheet of the sensor [51: BumbleBee2 Product Datasheet from PointGrey 2013], the sensor size is 157×47.4×36 mm with weight 342 grams. It provides two images from left and right CCDs with 43 degrees horizontal field of view, and then calculates the disparity using right image as target image plane. Therefore the coordinate of disparity map is the same

as right image plane. The total data acquiring frame rate is about 20Hz for a personal computer with i7-950 CPU or 10Hz for a laptop with Intel P8400. Figure 6.1(c)-(f) shows the stereo data with RGB Images from left and right CCD and corresponding disparity image calculated from RGB images. The intensity of the disparity is represented as gray image, that is, the brighter of the pixel, the higher the disparity value is. Note that the white area is invalid pixel with value 256. Since distance is inversely proportional to the disparity, in other words, the brighter of the pixel, the closer to the camera coordinate. Also note that the value of invalid pixel in depth map is 0.

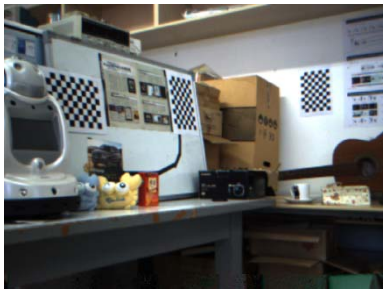
In this thesis, stereo vision data is captured by utilizing the two software development kit libraries of Bumblebee stereo vision camera, which are FlyCapture and Triclops SDK. The FlyCapture library uses IEEE 1394a bus as a communication interface to exchange information between computer and camera. The Triclops library uses the sum of absolute difference (SAD) correspondence method to estimate the disparity between two images captured by the stereo vision camera. To connect the Bumblebee stereo camera, the computer must have IEEE 1394a interface. In our platform, a personal computer is equipped with Uptech DV/1394 I/O Card. For an autonomous robot controlled by a laptop, it should be equipped with Uptech UTE 120 Combo Card to the ExpressCard slot and an external 12V power to drive the Bumblebee stereo camera.

(a)

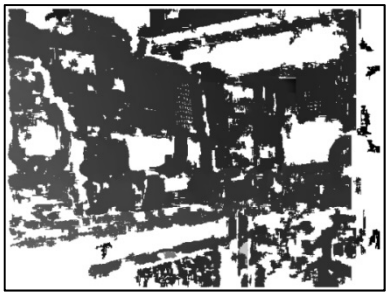
(b)



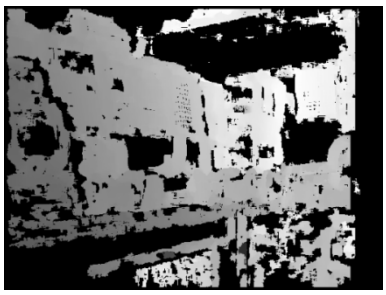
(c)



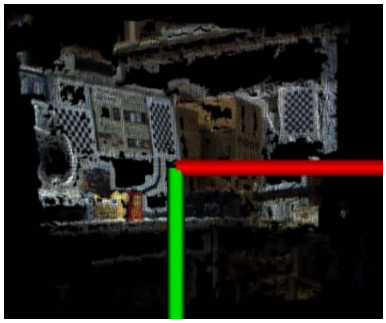
(d)



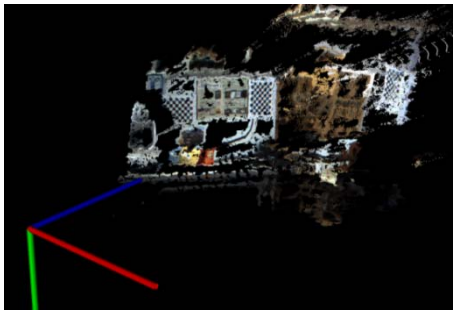
(e)



(f)



(g)



(h)

Figure 6.1: A brief introduction of Bumblebee2 BB2-03S2-60 stereo camera.

(a) Point Grey Bumblebee2 BB2-03S2-60. [51: BumbleBee2 Product Datasheet from PointGrey 2013]

(b) Data acquiring from the scenario for example.

(c)-(d) Image from left and right CCD respectively.

(e) The disparity map estimated by (c)-(d). Note that (d) is the target image in Bumblebee2 System.

(f) The depth map transformed from (e) with the relation $z(i, j) = fB / d(i, j)$.

(g)-(h) Point clouds generated from (e) and plotted on to 3D coordinate.

(a)

(b)

Figure 6.2: IEEE 1394 Interface

(a) Uptech DV/1394 I/O Card.

(b) Uptech UTE 120 Combo Card [50: UTE120 Combo ExpressCard from Uptech 2013].

Table 6.1: The specification of stereo camera BB2-03S2-60

Baseline	0.12 m
Focal Length	6mm
Horizontal Field of View (HFOV)	43 degrees
Image Resolution	Maximum for 640×480 pixels
CCD Frame Rate	48 Fps
Accuracy	$\Delta Z = \left\| \frac{fB}{d} - \frac{fB}{d+e} \right\|$, *where* $e = matching\ error$

Laser Range Finder

In order to evaluate the experimental result, two Hokuyo URG-04LX-UG01 laser range finders are used to be a benchmark. Hokuyo URG-04LX-UG01 could acquire the laser range in 240° with detection range about 20 mm – 5600 mm. The specification is listed in Table 6.2.

Hokuyo URG-04LX-UG01 laser range finder is capable of detecting in the range 20 mm – 5600 mm. However, for evaluating the object detection and tracking task in

long range up to 10 meter, SICK LMS100 laser range finder is used. SICK LMS100 could acquire the laser range in 270° with detection range about 0.5 m – 20 m, which is suitable for evaluating the proposed method. The specification is listed in Table 6.3.

(a) (b)

Figure 6.3: Hokuyo URG-04LX-UG01 and SICK LMS100 laser range finders.
(a) Appearance of URG-04LX-UG01 laser range finder [52: URG-04LX-UG01 from Hokuyo].
(b) Appearance of SICK LMS100 laser range finder [53: SICK LMS100 from SICK].

Table 6.2: The specification of Hokuyo URG-04LX-UG01

Detection Angle Range	240 Degrees (-30° *to* 210°)
Detection Range	20 mm – 4000 mm
Scanning Rate	10 Hz
Angular Resolution	0.36 Degree
Accuracy	$0.06-1m:\ \pm 30mm,\ \ 1-4m:\ \pm 3\%$

Table 6.3: The specification of SICK LMS100

Detection Angle Range	270 Degrees (-45° *to* 225°)
Detection Range	0.5 - 20 m
Scanning Rate	10 Hz
Angular Resolution	0.25 Degree
Accuracy	$\pm 30-40$ mm

6.2 Stereo Camera Localization and Mapping

To evaluate the overall 3D environment reconstruction system, several experiments are built to test each subsystem. To check the feature-based localization method, stereo camera capture a sequence of frame data in an indoor scenario with “L” shape with two laser range finders which are used to be a benchmark. Subsection 6.2.1 shows the experiment scenario and the construction of the platform. Subsection 6.2.2 shows the benefit of using RANSAC outlier rejection algorithm in the localization task. Subsection 6.2.3 shows the relation between localization accuracy and mapping quality from 2D laser data mapping result. Subsection 6.2.4 shows the accuracy of the feature-based localization result comparing to two laser range finders. Finally, Subsection 6.2.5 shows the 3D environment reconstruction result rendered in 3D window.

6.2.1 Experimental Scenario Setup

In this experiment, the scenario is constructed with size $2\times2\times2m^3$ as shown in Figure 6.4. To evaluate the localization method, the stereo camera is mounted with two Hokuyo URG-04LX-UG01 laser scanners orthogonally as shown in Figure 6.5(a). The stereo camera moves by the given commands according to the grid sheet with resolution $1\times1\ cm^2$ on the ground, shown in Figure 6.5(b) and (c). In addition, the pan angle is

given by the angle gage of the cradle head of Coman JS-4254+CV-0 tripod, as shown in Figure 6.5(d). The camera trajectory is composed by four paths as shown in Figure 6.6:

1. **Frame 0-8**: camera moves lateral by $+0.1m$ per step in the first path.
2. **Frame 9**: in the second path, camera rises $+0.085m$.
3. **Frame 10-15**: camera then rotates 15° per step until it reaches 90° in the third path.
4. **Frame 16-18**: for the final path, camera moves lateral by $+0.1m$ per step again.

The bottom-right image in Figure 6.6 illustrates the camera trajectory mentioned above, while the four image groups are the image captured from right CCD of the stereo camera corresponding to the four paths.

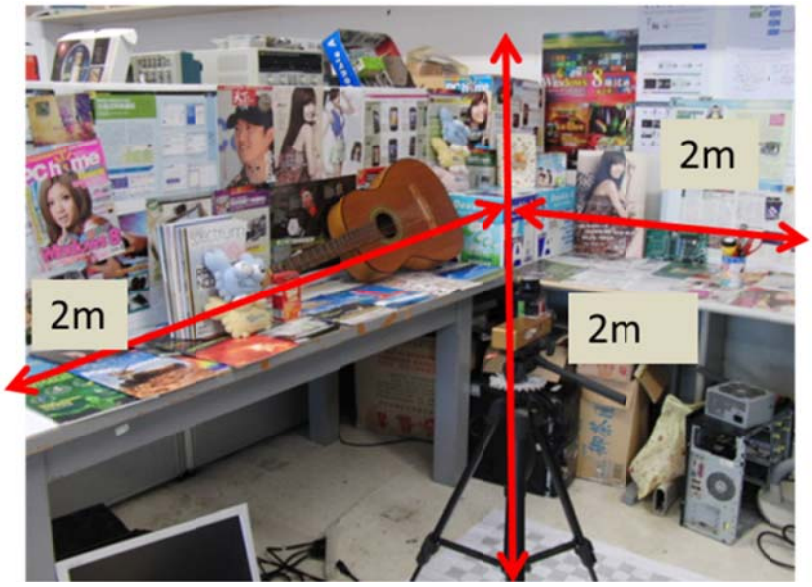
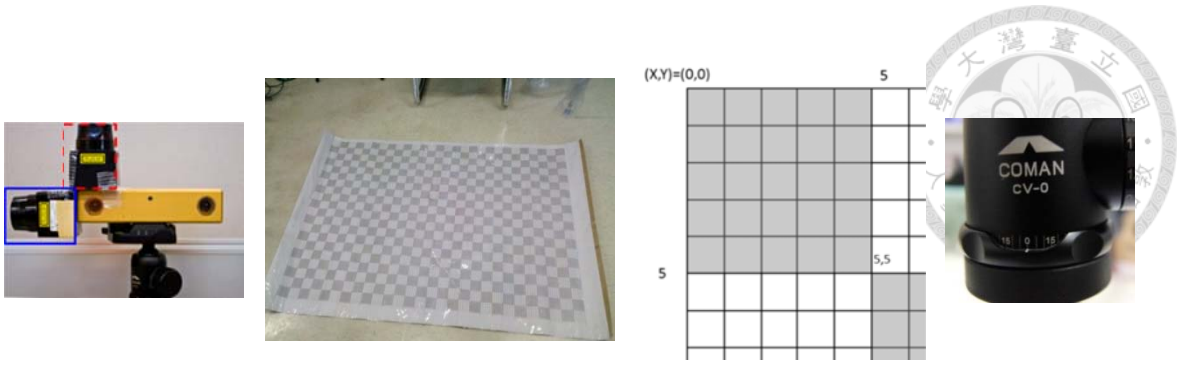


Figure 6.4: Experiment scenario



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

Figure 6.5: Experiment platform and accessories.

(a) Stereo camera with two HOKUYO laser scanners mounted orthogonally.

(b)(c) The grid sheet with resolution $1\times1\ cm^2$.

(d) The cradle head of Coman JS-4254+CV-0 tripod.

Figure 6.6: Camera path and corresponding image captured from right CCD

The laser scanners are mounted orthogonally to estimate the relative horizontal and vertical motion of the stereo camera. For the laser mounted horizontally as shown in Figure 6.7, the camera horizontal motion (X-Y plane) is estimated by using ICP registration method. For the laser mounted vertically shown in Figure 6.8, the camera

vertical motion (Z axis) is estimated just by comparing the average laser measurements in time k and $k-1$.

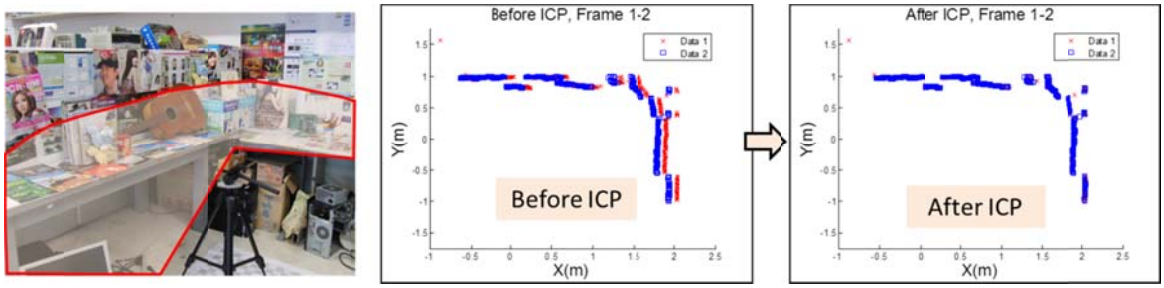
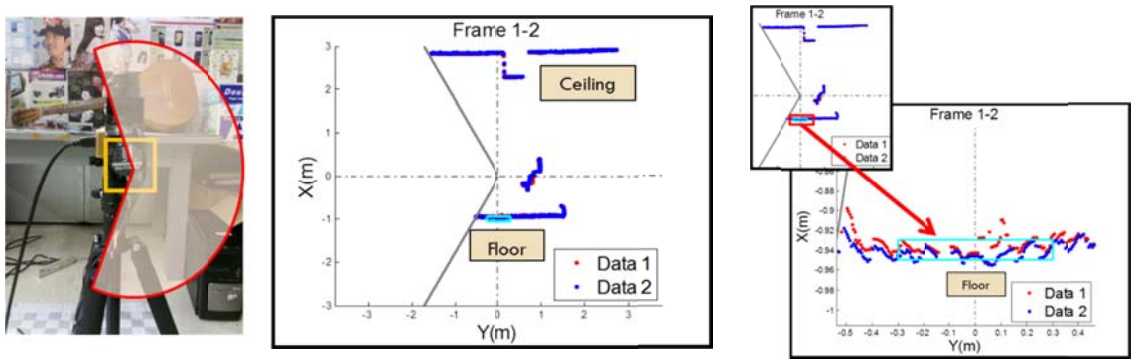


Figure 6.7: Horizontal laser data. The camera horizontal motion is estimated by applying ICP method to align two consecutive laser data.



(a) (b)

Figure 6.8: Vertical laser data

(a) Experiment scenario and corresponding vertical laser data.

(b) Laser height is determined by calculating the mean of the laser data from angle -5° to $+5^{\circ}$ and are specified a value with threshold at $0.94 \pm 0.01m$

6.2.2 The Effect of RANSAC Outlier Rejection Algorithm

As mentioned in Section 4.2, there have some wrong matching pairs between current and previous step. To show the benefit of using RANSAC outliers rejection algorithm, the localization with and without RANSAC algorithm results are plotted on the Figure 6.9 in bird's-eye view. The localization result with RANSAC outlier rejection algorithm is plotted as '-□-' colored in blue. The given command is plotted as '-∗-' colored in green, and the localization result estimated by laser using ICP method is plotted as '-×-' colored in red. The localization result which considering all the matching pairs to estimate camera position without using RANSAC outlier rejection is plotted as '-O-' colored in black. The camera position estimated by matching pairs without RANSAC rejection algorithm is not close to the results estimated by laser-ICP and the given command. On the other hand, the feature-based localization result with RANSAC algorithm is similar to the results estimated by laser-ICP and the given command. Therefore, it is obvious to show that some outliers will cause inaccurate localization result and can be solved by using RANSAC algorithm.

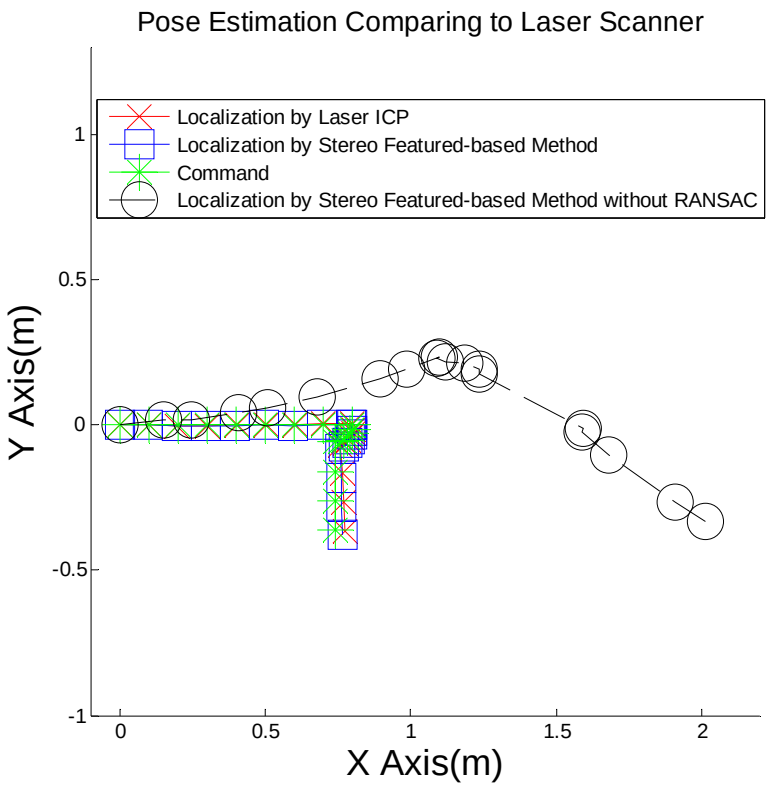
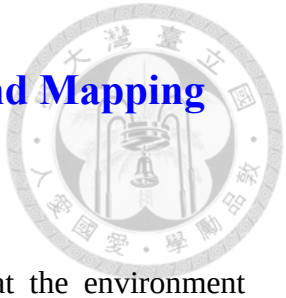


Figure 6.9: Comparing the result of using feature-based localization method with and without RANSAC outlier removal.



6.2.3 Relation between Localization Accuracy and Mapping Quality

The purpose of this experiment in this subsection shows that the environment reconstruction result is more accurate with better localization. Since only two laser scanners are mounted on the stereo rig, laser data can only have 5 degree of freedom. Thus, this experiment degenerates the problem dimension from 6-DoF to 3-DoF planar motion. The horizontal laser data is used to construct the 2-D top-view of the environment map. Laser map reconstruction results with different localization methods are shown in Figure 6.12, whereas Figure 6.10 demonstrates the localization results of three different approaches, and Figure 6.11 shows the translation displacement and rotation error of given commands and feature-based localization method comparing to laser-ICP. Blue square markers plotted in Figure 6.11(a) represent the translation displacement between feature-based localization method and laser-ICP, and blue bars plotted in Figure 6.11(b) show the rotation error of feature-based localization method comparing to laser-ICP; on the other hand, the green star signs indicate the translation displacement of localization result between given command and laser-ICP, and green bars plotted in Figure 6.11(b) show the rotation error of given commands comparing to laser-ICP. Figure 6.11 shows the feature-based localization result is better than the given commands.

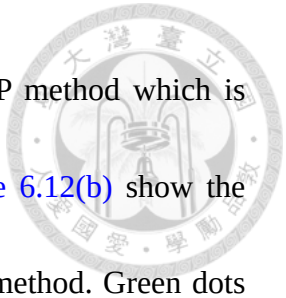


Figure 6.12(a) shows the laser map reconstructed by laser-ICP method which is used to be the experiment benchmark. Blue dots plotted on Figure 6.12(b) show the laser map reconstruction result by using feature-based localization method. Green dots plotted on Figure 6.12(c) show the laser map reconstruction result by the given commands. Almost blue dots in Figure 6.12(b) are overlapped to the red dots, but the green dots in Figure 6.12(c) has a slightly displacements. This shows that the laser data mapping result by feature-based localization method is better than the given commands, and therefore it gives the conclusion that the more accurate localization constructs the better mapping result. Note that in the third path at frame index #10-15, which the camera rotate 15° in each step, the translation and rotation errors of commands and feature-based localization method both rise up. This phenomenon shows that in the localization task, rotation motion is a challenge problem to be solved.

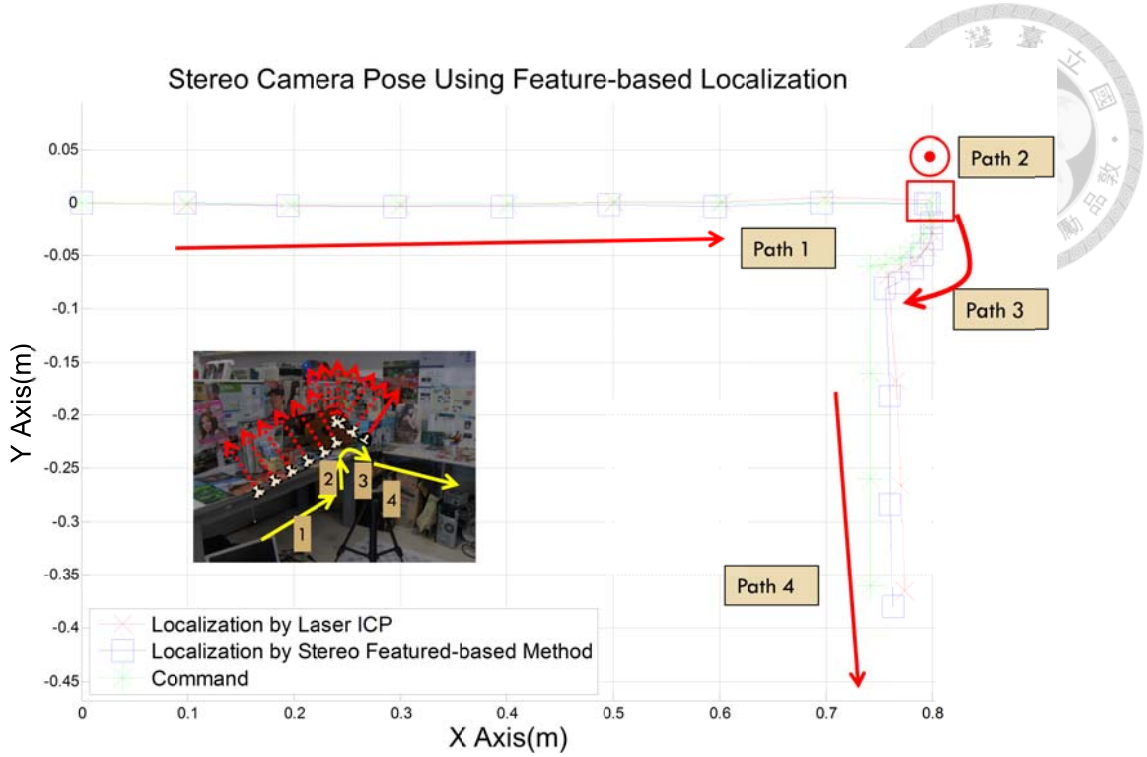
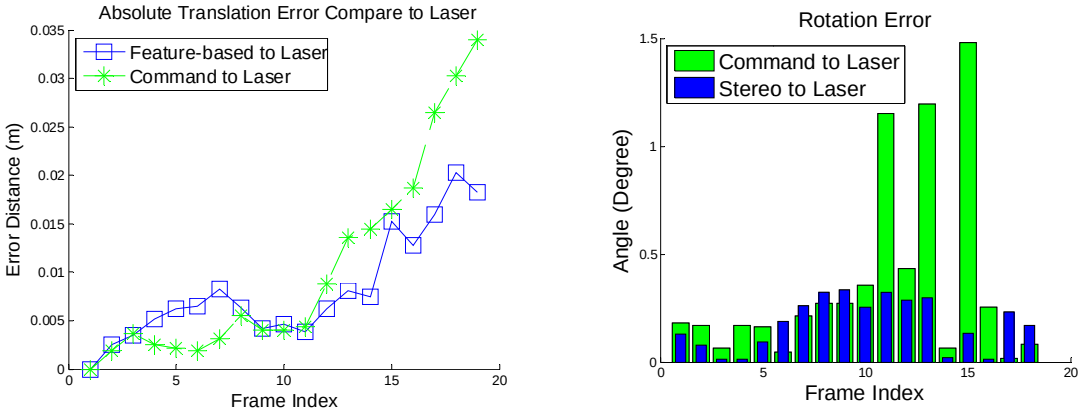


Figure 6.10: Top view of the camera path. Red cross signs represent the positions estimated by laser-ICP; blue square signs indicate the positions estimated by stereo camera feature-based localization method; green star signs show the given command positions.



(a) (b)

Figure 6.11: Translation and rotation error comparing to laser scanner

(a) The translation error (displacement) comparing to laser-ICP. Blue squares represent the error of the feature-based localization method, while the green stars indicate the error of the given commands.

(b) The rotation error comparing to laser-ICP. Blue bars show the error between feature-based localization and laser-ICP, while the green bars indicate the error between the given commands and laser-ICP.

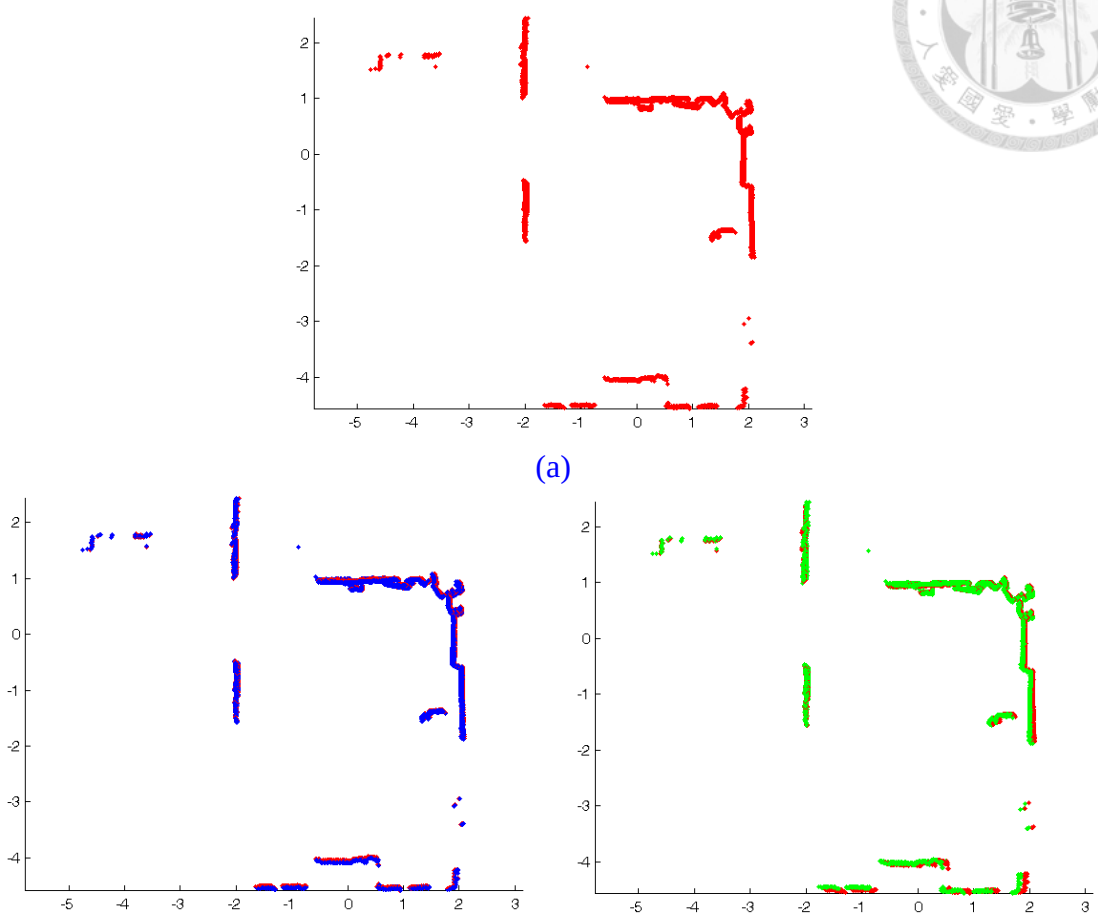


Figure 6.12: Laser data mapping with localization by stereo feature-based localization method and the given commands.

(a) Laser map constructed with laser-ICP localization method

(b) Laser map constructed with stereo vision feature-based localization method

(c) Laser map constructed with the given command

6.2.4 The Accuracy of Feature-based Localization Algorithm

In this subsection, the accuracy of the feature-based localization is analyzed by comparing to laser scanners. Each x-, y- and z-component of the platform positions which are given by different localization methods are plotted on to the left column of Figure 6.13, while the absolute components errors are on the right column. The blue square signs in Figure 6.13(a), (c) and (e) indicate the positions estimated by stereovision feature-based localization method, and the blue square signs in Figure 6.13(b), (d) and (f) show the corresponding absolute errors of each components comparing to laser scanner. From Figure 6.13(b), (d) and (f), each component error between feature-based localization and laser-ICP is less than 0.02 meter, and the absolute translation error is less than 0.025 meter as shown in Figure 6.14, where the absolute translation error is the Euclidean distance which is defined as follows:

$$E_{Translation}(i) = \sqrt{E_x(i)^2 + E_y(i)^2 + E_z(i)^2} \tag{6.1}$$

$$E_x(i) = x_{i,laser-ICP} - x_{i,Feature-based} \tag{6.2}$$

In addition, Figure 6.15(a) shows the accumulating moving distance in each step, which is defined as follows:

$$d(k) = \sum_{i=1}^{k} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{6.3}$$

$$d(0) = 0 \tag{6.4}$$

For moving 1.283 meter determined by laser, the accumulating moving distance drifts

of feature-based localization method is about 0.005 m, as shown in Figure 6.15(b). Both the translation errors and accumulating moving distance drifts are in a small range, which can be considered as the laser sensing uncertainty.

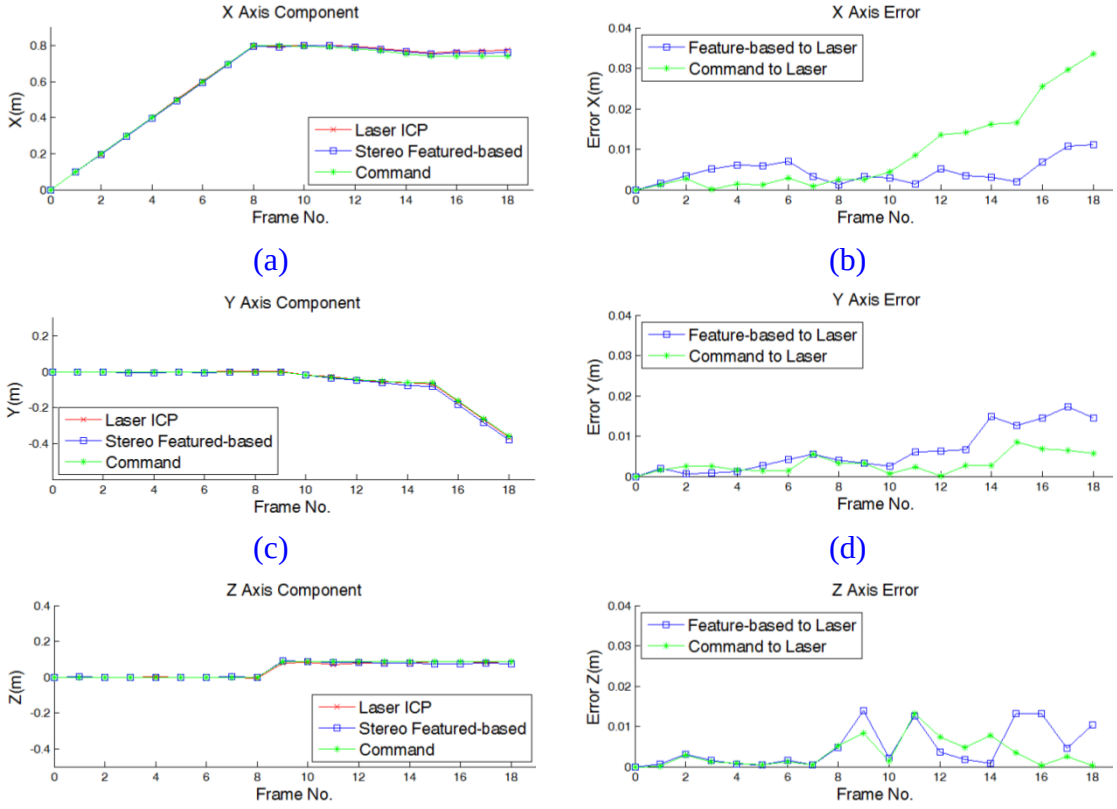


Figure 6.13: Stereo featured-based localization comparing to laser scanners.

(a)(c)(e) The X-, Y- and Z-components of the positions given by three different approaches. Red cross signs indicate the platform positions given by the laser-ICP method, blue square signs represent the positions given by stereo camera feature-based localization approach, and green star signs express the positions provided by the given commands.

(b)(d)(f) The errors of each components comparing to laser range finder. Blue square signs represent the error between feature-based localization approach and laser-ICP; green star signs express error between commands and laser-ICP.

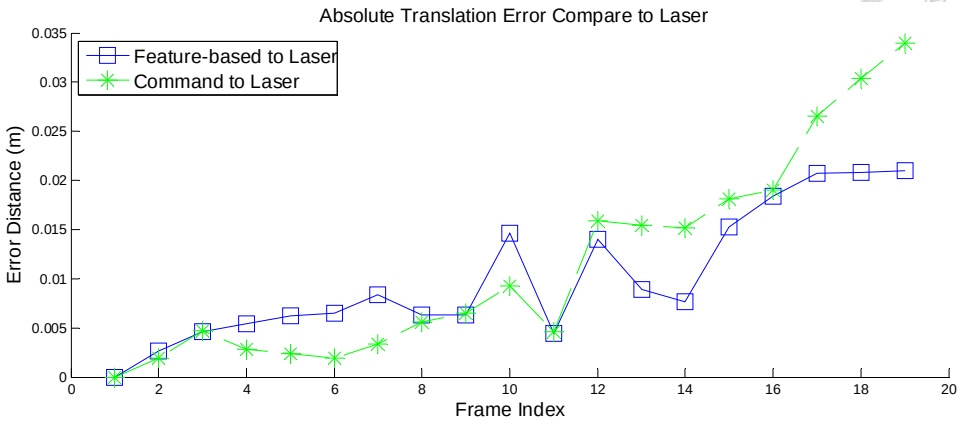
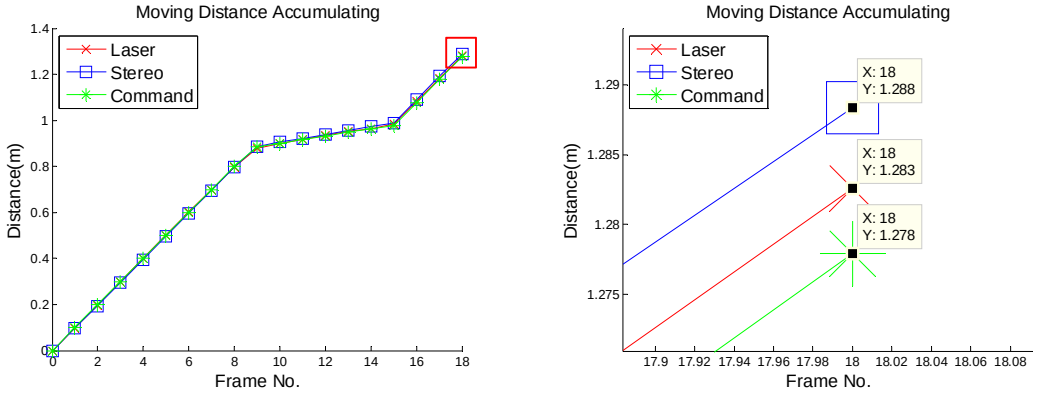


Figure 6.14: Absolute translation error comparing to laser range finder.



(a) (b)

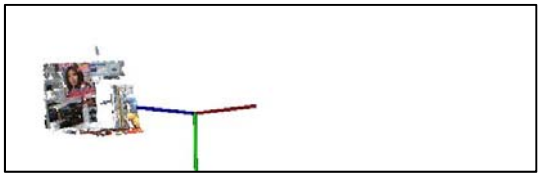
Figure 6.15: Accumulate distance for moving 18 steps.

(a) The global view of the distance accumulation

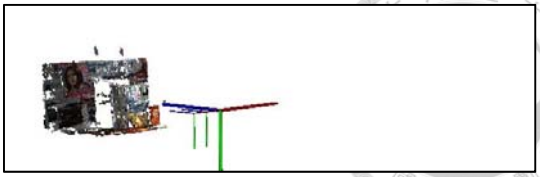
(b) The local view of 18-th distance accumulation.

6.2.5 Three-Dimensional Reconstruction

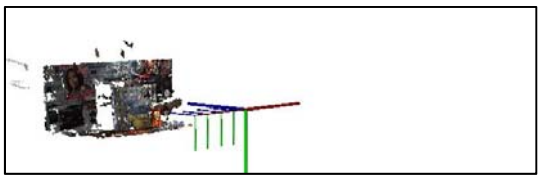
The stereo vision measurements with color and spatial information (RGBXYZ data structure) are plotted on to world coordinate using Point Cloud Library in C++ code [54: Point Cloud Library from PCL Website 2013], as shown in Figure 6.16, Figure 6.17 and Figure 6.18. The red, green and blue sticks which are placed orthogonally represent the camera coordinate in each step. Figure 6.16 demonstrates the mapping results in each step with two time interval. Points in brighter color represent the measurement points in current step, while the other points in darker color indicate points of the global map. Figure 6.17(a) shows the experiment scenario, Figure 6.17(b) shows the reconstruction result by the proposed method, and Figure 6.17(c) shows the reconstruction result by given commands. Figure 6.18 shows the local views of the reconstruction result to demonstrate the better mapping result by applying feature-based localization method than the given commands. It is obvious to see that some object points do not align together in the 3D model reconstructed by the given commands shown in the Figure 6.18(c), while the mapping result reconstructed by the feature-based localization method is clearer, as shown in Figure 6.18(b). This is because that the localization result by stereo vision localization method is more accurate than the given command, as mentioned in Subsection 6.2.3 and 6.2.4. To evaluate the mapping quality by quantitative analysis, peak-signal-to-noise ratio (PSNR) is used in the Subsection 6.2.6.



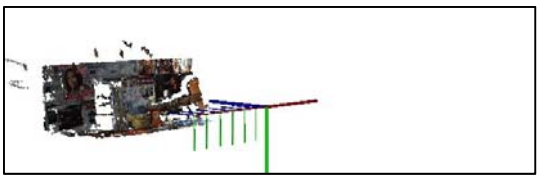
(a) step 0



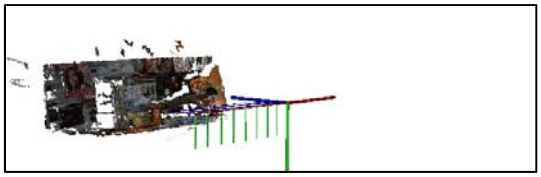
(b) step 2



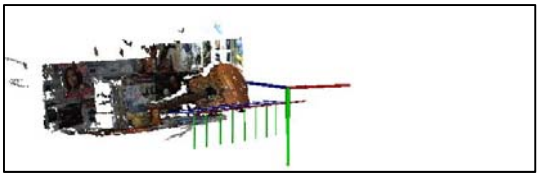
(c) step 4



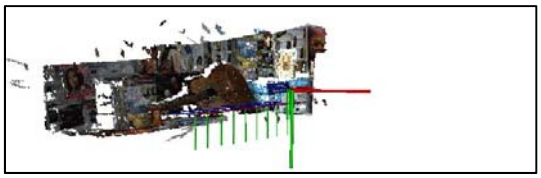
(d) step 6



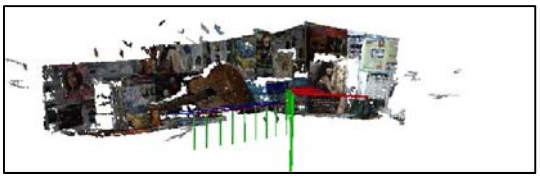
(e) step 8



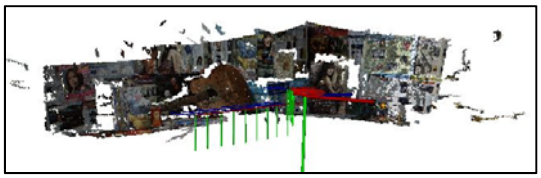
(f) step 10



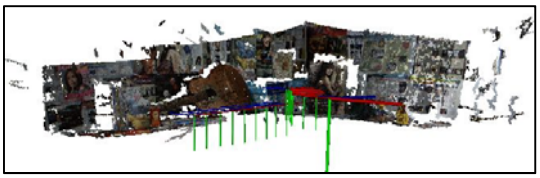
(g) step 12



(h) step 14



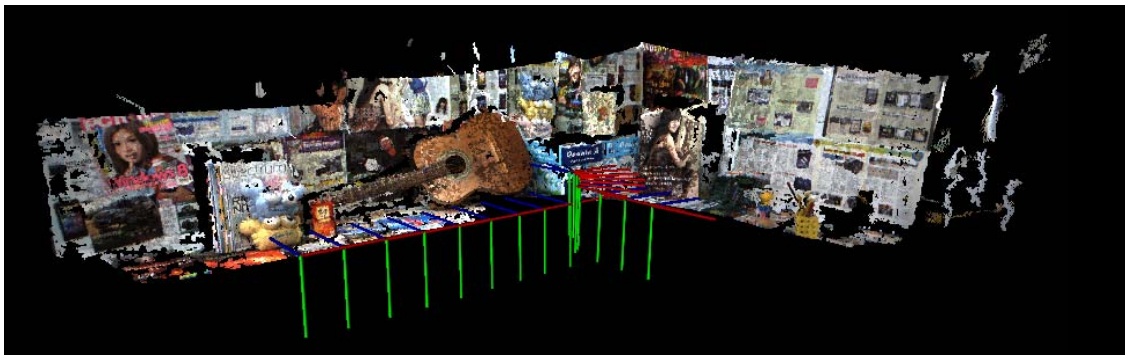
(i) step 16



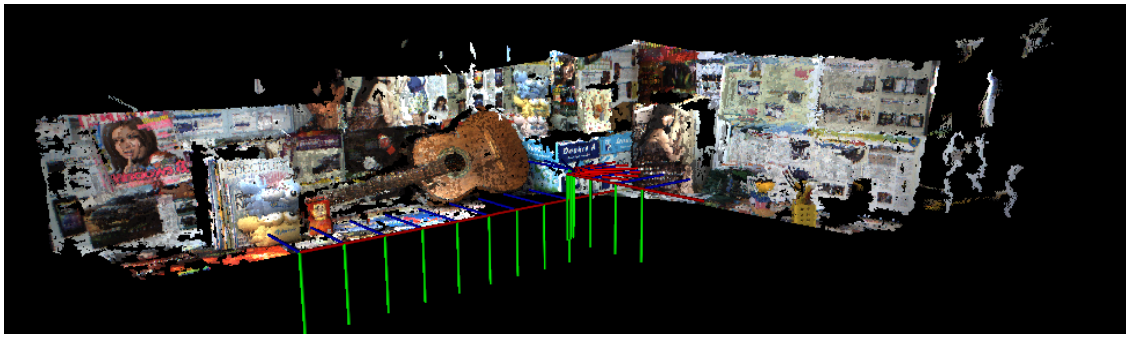
(j) step 18

Figure 6.16: The mapping results in each step with two time interval.

(a)



(b)



(c)

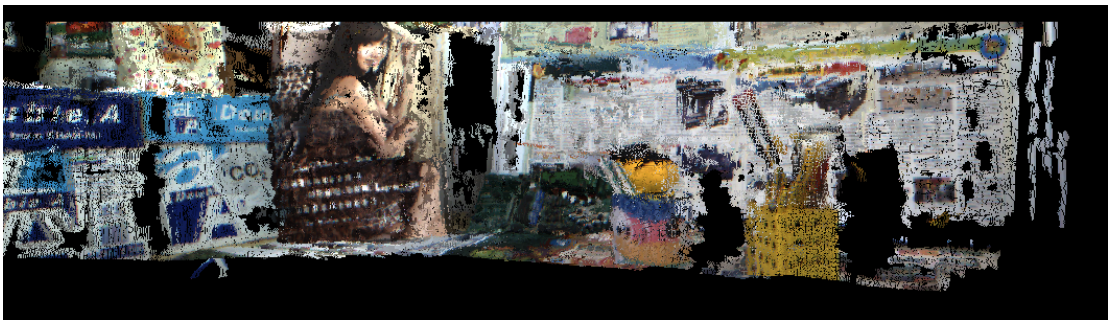
Figure 6.17: Result of the three-dimensional environment

(a) The experiment scenario

(b) Environment reconstruction using stereo camera data with feature-based localization method.

(c) Reconstruction result by the given commands.

(a)



(b)

Figure 6.18: The local views of the reconstruction results at the same camera viewpoint.

(a) Reconstruction result by feature-based localization method.

(b) Reconstruction result by the given commands. It is obvious to see that some object points do not align together.

6.2.6 Mapping Quality and the Proposed Stereo Refinement Algorithm Evaluation

In the previous subsection, the 3D environment reconstruction results of different localization approaches are shown in the 3D window. Although the mapping qualities have been compared by qualitative analysis in Subsection 6.2.5, the quantitative analysis is necessary to analyze the mapping result. On the other hand, the proposed stereo refinement algorithm can be evaluated in the same manner since the goal of the algorithm is to improve the mapping quality. Therefore, the mapping quality and stereo refinement algorithm is evaluated together in the visual aspect in this subsection. Moreover, to evaluate the accuracy of the proposed stereo refinement algorithm, another experiment is built in the spatial aspect.

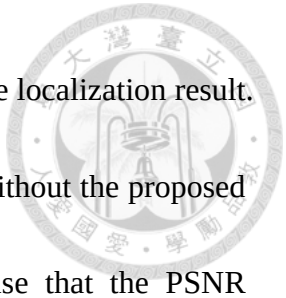
First, for visual aspect, the 3D reconstruction model is evaluated by comparing the image projected from the 3D model to target image in each step. Each projection image is built by projecting the 3D model points to image plane according to the camera pose using pin-hole camera model, which is mentioned in Section 3.1 and illustrated in Figure 6.19. Figure 6.19(b) is an example of the projection image result acquired from the 3D model. To comparing the projection image to target image, peak signal-to-noise ratio (PSNR) is used as a similarity index. PSNR is defined as follows [61: Szeliski 2010]:

$$PSNR = 20\log_{10}\left(\frac{I_{\max}}{RMS}\right) \tag{6.5}$$

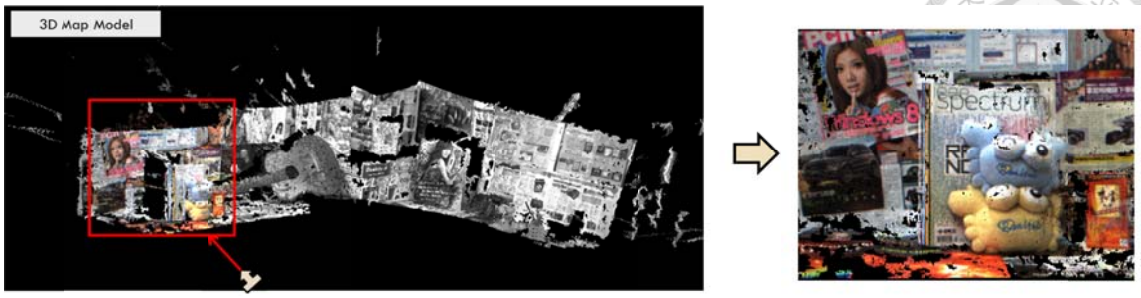
$$RMS = \sqrt{\frac{1}{n}\sum_{x}\left[I(\boldsymbol{x}) - \hat{I}(\boldsymbol{x})\right]^2} \tag{6.6}$$

where $I_{\max} = 255$ the maximum intensity of the image is, RMS is the root mean square of the image, $I(\boldsymbol{x})$ is the target image as benchmark and $\hat{I}(\boldsymbol{x})$ is the projection image in this experiment. Note that not every pixel has projection value due to no point can be projected from 3D model and the digitalization effect, as shown in Figure 6.22(a), the invalid pixels are not counted in calculating the PSNR value.

Figure 6.20(a) is the target image used as a benchmark, Figure 6.20(b) shows the projection image from the 3D model reconstructed by the given commands as shown in Figure 6.17(c), Figure 6.20(c) shows the projection image from the 3D model reconstructed by stereo vision feature-based localization method, and Figure 6.20(d) represents the project image from the 3D model reconstructed by feature-based localization with the proposed stereo refinement method. The PSNR values are listed in Table 6.4 and plotted in Figure 6.21(a). It can be observed from Figure 6.21(a) that all the PSNR values of the 3D model reconstructed by the given commands are lower than others, which are the PSNR values of the 3D model reconstructed by feature-based localization method. In addition, as mentioned previously, the localization result of the given commands is inaccurate than the feature-based localization method. Therefore,



this means that the better 3D model is reconstructed by more accurate localization result. The PSNRs of 3D model from feature-based localization with and without the proposed stereo refinement method are close to each other. This is because that the PSNR consider the average difference of each pixel from testing to target image, and the proposed stereo refinement algorithm do increase the number of valid pixels but do not improves the image quality. Then, the percentages of the number of the valid pixels projected from 3D model with and without stereo refinement method are listed in Table 6.4 and plotted on Figure 6.21(b), the black solid lines with cross sign represent the result with applying stereo refinement method, while the red dash dot line with square sign express result without applying stereo refinement. It can be observed that all the percentages of the result with applying stereo refinement method are larger than the result without applying. The blue dash lines with star sign indicate the increasing percentages of each frame data, and the mean of the increasing percentages is 3.08% with standard deviation 1.09% according to the right column of Table 6.4. This shows that the proposed refinement method can increase the data number of the 3D model by 3.08% without decreasing PSNR which represents the model quality. Thus, with the proposed stereo refinement method, the 3D model is better than others.



(a) (b)

Figure 6.19: The 3D model projects to image plane with certain camera poses

(a) The region enclosed by the red rectangle is projected to the image plane by pin-hole camera model.

(b) The result of projecting the 3D model to the image plane.

(a) (b)

(c) (d)

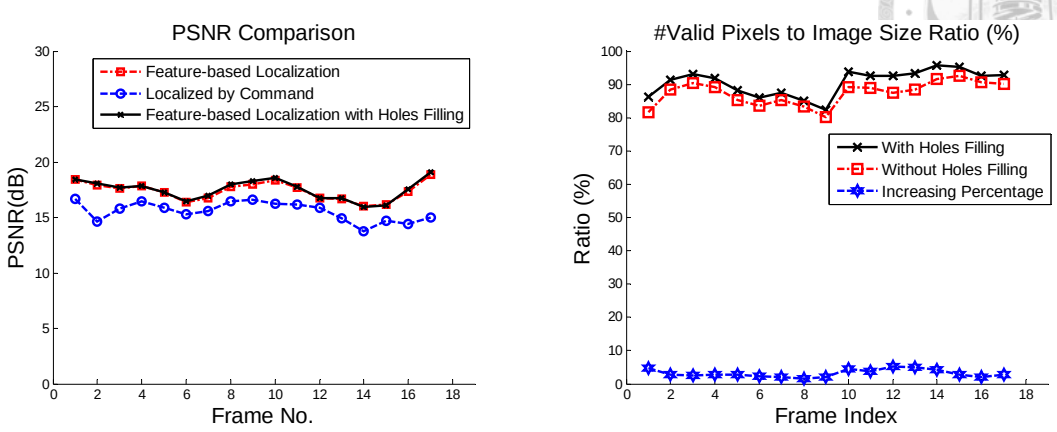
Figure 6.20: Illustration of the concept of using PSNR as similarity index to compare the 3D reconstruction quality in the color space viewpoint.

(a) The target image as ground truth.

(b) Image projected from 3D model reconstructed by given command.

(c) Image projected from 3D model reconstructed by proposed localization method.

(d) Image projected from 3D model reconstructed by proposed localization method with proposed stereo refinement method.

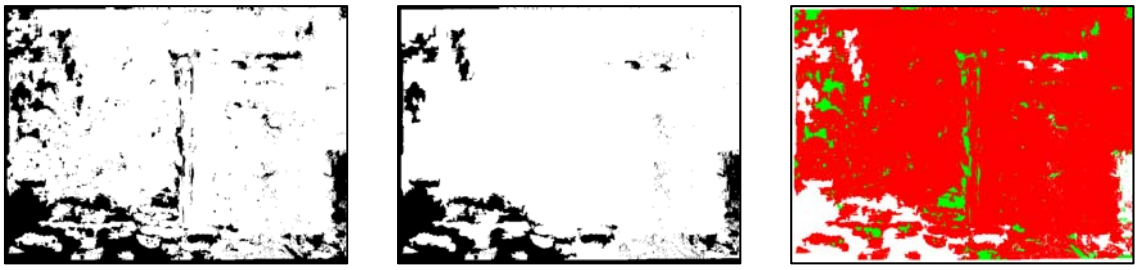


(a) (b)

Figure 6.21: Comparing the 3D reconstruction result in different case.

(a) Comparing each case to target image using PSNR value.

(b) Number of pixels projected from 3D model.



(a) (b) (c)

Figure 6.22: Valid pixels projected from 3D model with and without applying the proposed stereo refinement method. White area in (a) and (b) indicate the valid pixel, while black region represent the pixels without valid value.

(a) The valid pixels projected from 3D model without any data processing.

(b) The valid pixels projected from 3D model with applying the proposed stereo refinement method.

(c) Indicating each pixel is in state valid, invalid or refinement. Points in white represent the invalid pixels, the red and green region indicate the valid points, while the green areas are also considered as the pixels refined by the proposed method.

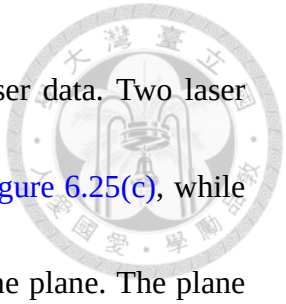
Table 6.4: 3D model projected to image plane and compare to target image.

	With Stereo Refinement		Without Stereo Refinement		
Items	PSNR	# Valid Pixels (%)	PSNR	# Valid Pixels (%)	# Valid Pixels Increment (%)
Frame 1	17.97	280307 (91.25%)	17.88	271442 (88.36%)	2.89%
Frame 2	17.71	285761 (93.02%)	17.69	278069 (90.51%)	2.51%
Frame 3	17.90	281997 (91.80%)	17.95	273734 (89.11%)	2.69%
Frame 4	17.29	271269 (88.30%)	17.37	262360 (85.40%)	2.90%
Frame 5	16.47	264272 (86.03%)	16.39	257095 (83.69%)	2.34%
Frame 6	16.86	268400 (87.37%)	16.75	262062 (85.31%)	2.06%
Frame 7	17.98	261130 (85.00%)	17.89	256217 (83.40%)	1.60%
Frame 8	18.58	252685 (82.25%)	18.24	246213 (80.15%)	2.10%
Frame 9	19.28	288396 (93.88%)	18.76	274228 (89.27%)	4.61%
Frame 10	18.80	284476 (92.60%)	18.43	272929 (88.84%)	3.76%
Frame 11	18.37	284750 (92.69%)	18.16	268865 (87.52%)	5.17%
Frame 12	17.89	286571 (93.28%)	17.63	271339 (88.33%)	4.95%
Frame 13	16.46	294475 (95.86%)	16.47	281228 (91.55%)	4.30%
Frame 14	16.45	292683 (95.27%)	16.44	284420 (92.58%)	2.70%
Frame 15	17.50	284657 (92.66%)	17.29	278641 (90.70%)	1.96%
Frame 16	19.52	285531 (92.94%)	19.28	276996 (90.17%)	2.77%
Frame 17	20.18	284094 (92.48%)	19.63	274849 (89.47%)	2.99%
Mean	17.9535	279497 (90.98%)	17.7794	270040 (87.90%)	3.08%
STD	1.0913	11697 (3.81%)	0.9489	10110 (3.29%)	1.09%

6.2.7 Evaluate the Proposed Stereo Refinement in Spatial Aspect

In Subsection 6.2.6 the proposed stereo refinement method is evaluated in the visual aspect using PSNR to compare the image projected from 3D model to target image. This evaluation method considers the local appearance of the 3D model by image-based approach. However, it does not evaluate the accuracy of the missing data (hole) filling of the proposed stereo data refinement in spatial aspect. Thus, this subsection focuses on evaluating the proposed stereo refinement algorithm in spatial aspect by constructing another experimental scenario.

In this experiment, a plane is placed in front of the stereo camera rig to be the benchmark, as shown in Figure 6.23. The plane is measured by using two orthogonal laser scanners and its plane parameters in stereo camera coordinate are estimated by using these local laser data. The plane is located at different positions in order to be measured by stereo camera rig in different viewing angles. Five viewing angles are set, that is, $-45^{\circ}, -30^{\circ}, 0^{\circ}, 30^{\circ}$ and 45° respectively, which are shown in Figure 6.23(c)-(g). To illustrate the evaluation method conveniently, Data #3 is used as an example, which the viewing angle is 0°. The plane is measured by the stereo camera and laser range finders as shown in Figure 6.24(a). Figure 6.25 shows the concept of estimating the plane parameters in the camera coordinate of Data #3. Figure 6.25(a) shows the



horizontal laser data, whereas Figure 6.25(b) shows the vertical laser data. Two laser data are transformed to the stereo camera coordinate as shown in Figure 6.25(c), while the data pointed by the green arrow indicate the measurements on the plane. The plane parameters are estimated by using the least-square method with these local laser data, which are selected manually as shown in Figure 6.25(d). Figure 6.26 shows the result of plane parameters estimation by using the local laser data. To compare the interpolation result in the coordinate of the depth map, the plane depth map is built by inverse projecting the points on the plane to the image coordinate. According to the definition of a plane which is written as follows:

$$Ax + By + Cz = D \tag{6.7}$$

And considering pin-hole model as mentioned in Section 3.1, substituting the Equation (3.1)-(3.2) into Equation (6.7), the equation becomes:

$$A\left(\frac{uz}{f}\right) + B\left(\frac{vz}{f}\right) + Cz = D \;\Rightarrow\; z = \frac{fD}{Au + Bv + C/f} \tag{6.8}$$

For a certain pixel (u,v) on the image coordinate, its depth value becomes:

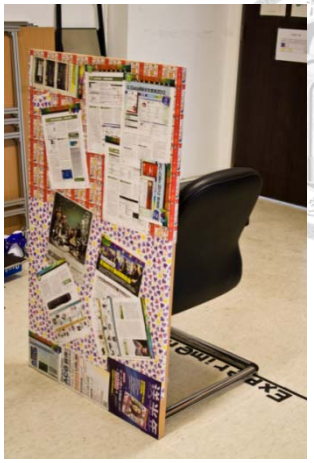
$$Depth(u,v) = z(u,v) = \frac{fD}{Au + Bv + C/f} \tag{6.9}$$

Therefore, the plane depth map can be built by using the Equation (6.9), and the construction result is shown in Figure 6.27(b).

The 200×200 rectangular region of interest (ROI) is selected at the image center to be the comparison area, as shown in Figure 6.28(a). The invalid pixels (missing data

areas) are filled by the proposed stereo refinement method with two different interpolation approaches and the results are shown in Figure 6.28(e) and (g), and the corresponding ROI patches are shown in Figure 6.28(f) and (h). The pixels in the selected ROI are compared to the same ROI patch of the plane depth map estimated from the laser data. Figure 6.28(d) shows the estimated plane depth map ROI patch of Figure 6.28(c). Since the processing pixels are the missing data area as the white region in Figure 6.28(i), only these pixels are compared to the estimated plane depth map. Figure 6.28(j) shows the absolute difference between the ROI patch of the proposed DOL interpolation approach and the ROI patch of the estimated plane depth map, while Figure 6.28(k) shows the absolute difference between the refinement result of RBF approach and the benchmark (Figure 6.28(d)). Figure 6.29(a) shows the absolute differences of each filling pixel, and Figure 6.29(b) shows the histogram of these absolute differences. For the five experimental datasets, each mean and standard deviation of the interpolation errors comparing to the estimated plane depth are listed in Table 6.5, and the processing times are listed in Table 6.6. It can be observed that the means and standard deviations of the interpolation result by DOL are all slightly lower than the result by RBF, which shows the better result in the planar case. Moreover, according to Table 6.6, the processing time of DOL is approximately lower three times than the processing time of RBF, which shows the better computation efficiency.

(a)



(b)

(c)

(d)

(e)

(f)

(g)

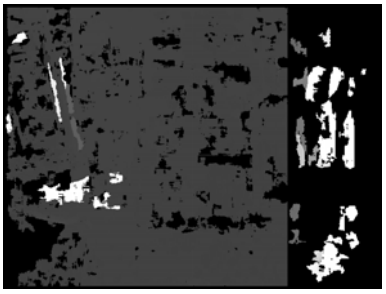
Figure 6.23: Experiment scenario setup. A plane stands in front of the stereo camera with different view angle.

(a)(b) A plane stands in front of the sensor platform.

(c)-(g) Measuring the plane with different angles, $-45^{\circ}, -30^{\circ}, 0^{\circ}, 30^{\circ}$ and 45° respectively.

(a)

(b)



(c)

Figure 6.24: The target image and the corresponding depth map of Data #3.

(a) The plane stands in front of the camera with viewing angle 0°.

(b) Target image from right CCD of the stereo camera.

(c) The corresponding depth map.

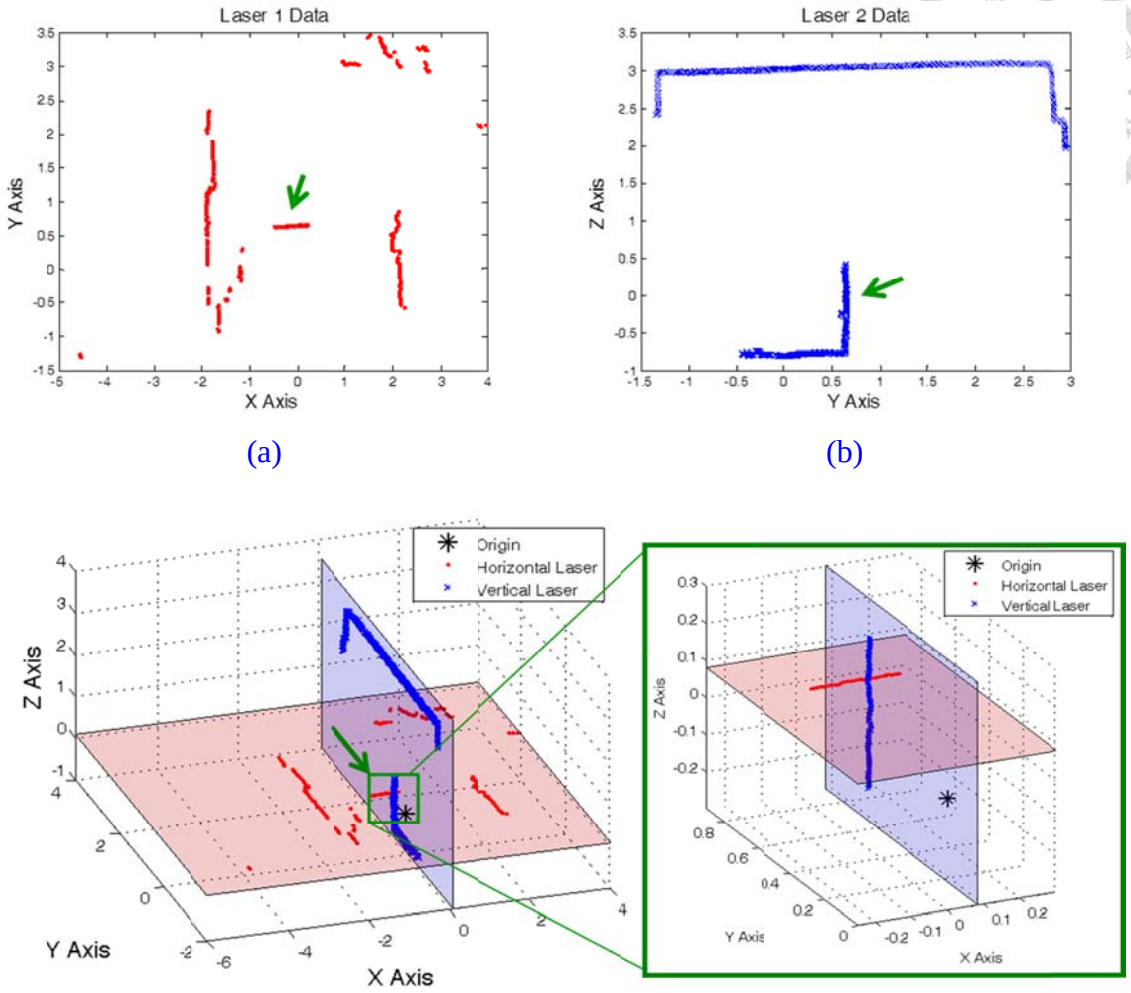


Figure 6.25: Two orthogonal laser data and transformed to camera coordinate. The local laser data is used to be the input of plane estimation.

(a) Data acquired from horizontal laser scanner.

(b) Data acquired from vertical laser scanner.

(c) Align two laser data to stereo camera coordinate with corresponding detection plane.

(d) Local data of the horizontal and vertical laser are selected manually to estimate the plane parameters.

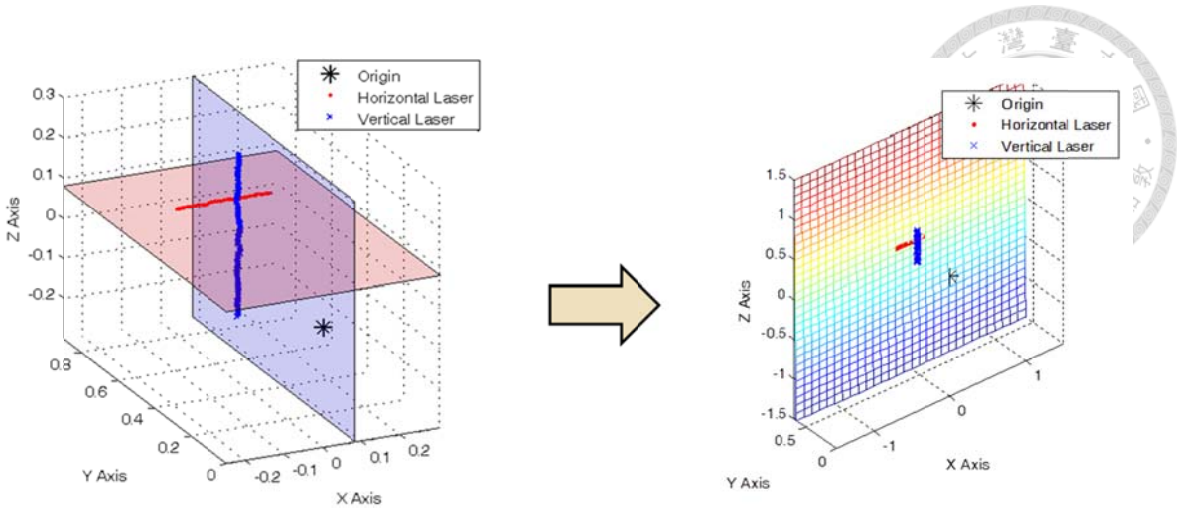
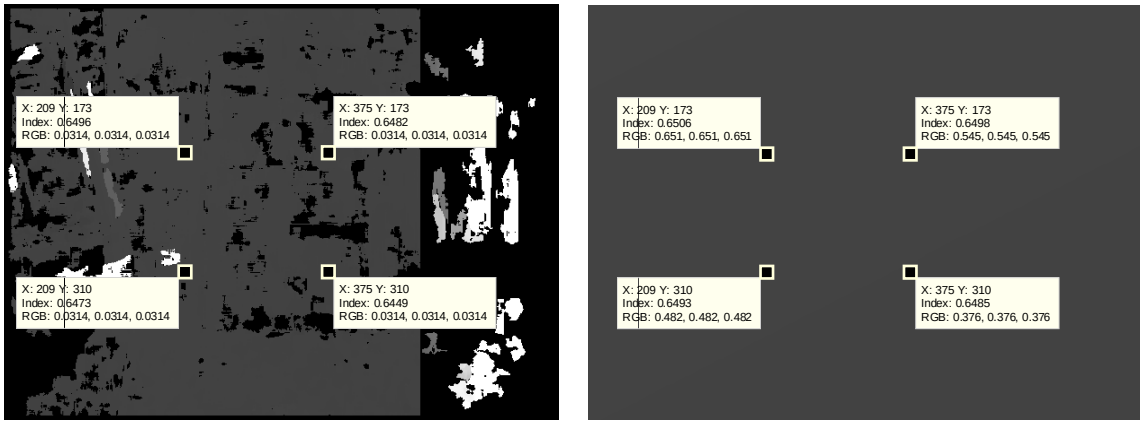


Figure 6.26: Using local laser data to estimate the plane parameters

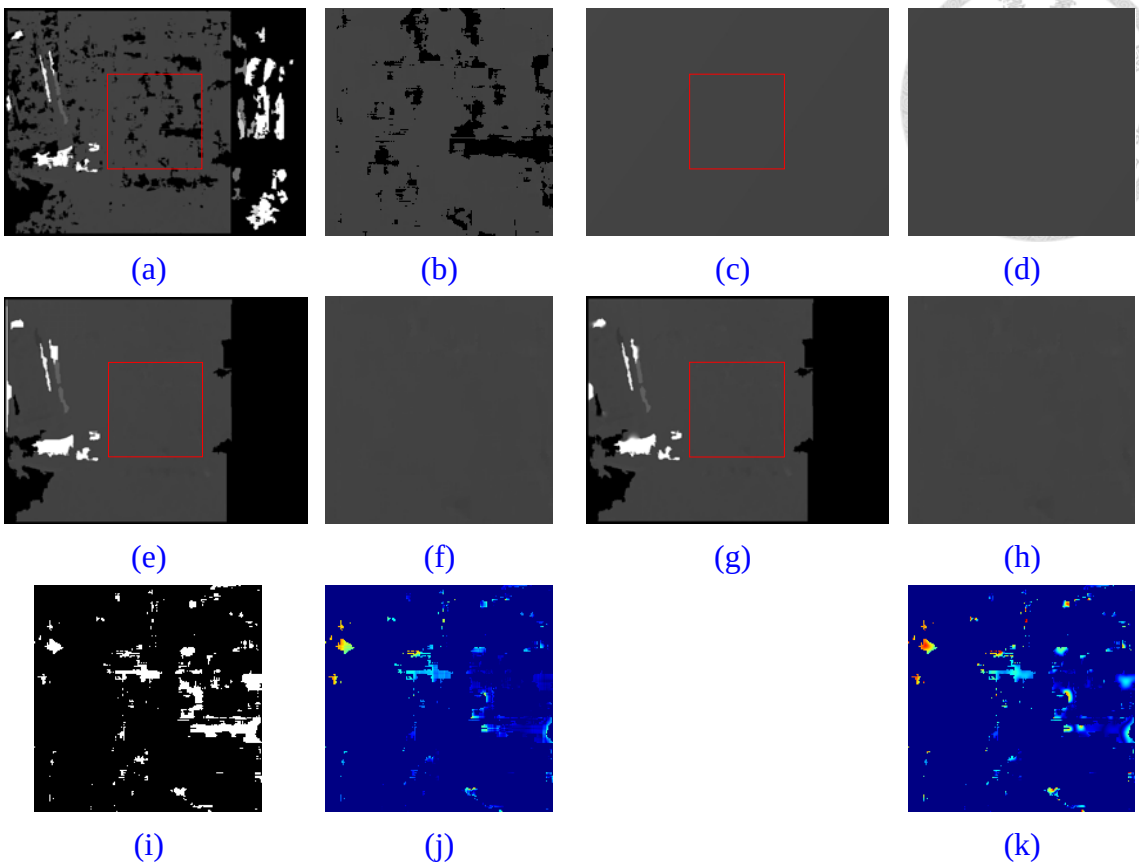


(a) (b)

Figure 6.27: Comparing raw depth map and the depth map generated from the plane.

(a) Raw depth map from stereo camera

(b) Depth map generated from the plane estimated by local orthogonal laser data.



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

(e) (f) (g) (h)

(i) (j) (k)

Figure 6.28: The 200×200 rectangular ROI is selected, which is enclosed as in the depth maps.

(a)(b) The raw depth map and the corresponding ROI patch.

(c)(d) The depth map estimated from laser data and the corresponding ROI patch.

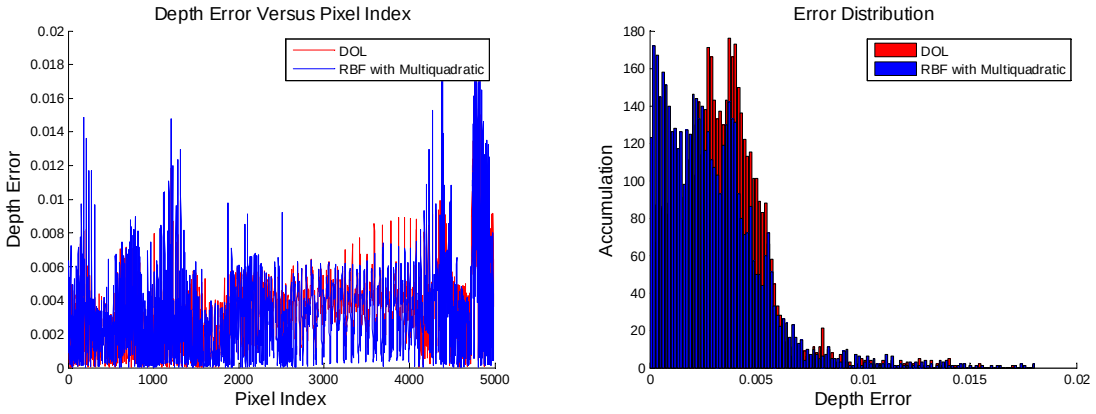
(e)(f) The stereo refinement result of dual orthogonal linear (DOL) interpolation and the corresponding ROI patch.

(g)(h) The stereo refinement result of radial basis function (RBF) interpolation and the corresponding ROI patch.

(i) The white areas indicate the region of missing data (hole) to be filled.

(j) The absolute difference of the filling region between (d) and (f).

(j) The absolute difference of the filling region between (d) and (h).



(a) (b)

Figure 6.29: The absolute error between the depth of interpolating pixels and the depth generated from laser data. Red lines represent the result of the dual orthogonal linear interpolation approach, while the blue lines indicate the result of radial basis function method.

(a) Pixel index versus depth error.

(b) Histogram of depth error.

Table 6.5: The mean and standard deviation of interpolation errors comparing to laser scanner (m).

Data Index	Data #1	Data #2	Data #3	Data #4	Data #5
Angle	–45°	–30°	0°	30°	45°
$\overline{Error}_{DOL}$	0.0033	0.0037	0.0029	0.0125	0.0235
σ_{DOL}	0.0033	0.0034	0.0021	0.0095	0.0169
$\overline{Error}_{RBF}$	0.0051	0.0051	0.0028	0.0145	0.0253
σ_{RBF}	0.0046	0.0041	0.0025	0.0106	0.0181

Table 6.6: Comparing the processing time with different interpolation approaches.

Data Index	Data #1	Data #2	Data #3	Data #4	Data #5
Angle	–45°	–30°	0°	30°	45°
Average processing time (s)					
DOL	0.8206	1.2909	1.4425	1.1886	1.1476
RBF	2.2014	2.9561	3.8157	3.0266	3.0398
Number of the interpolation pixels (N)					
DOL & *RBF*	15784	19161	24636	16614	16577
Average processing time per pixel (ms/N)					
DOL	0.0520	0.0674	0.0586	0.0715	0.0692
RBF	0.1395	0.1543	0.1549	0.1822	0.1834

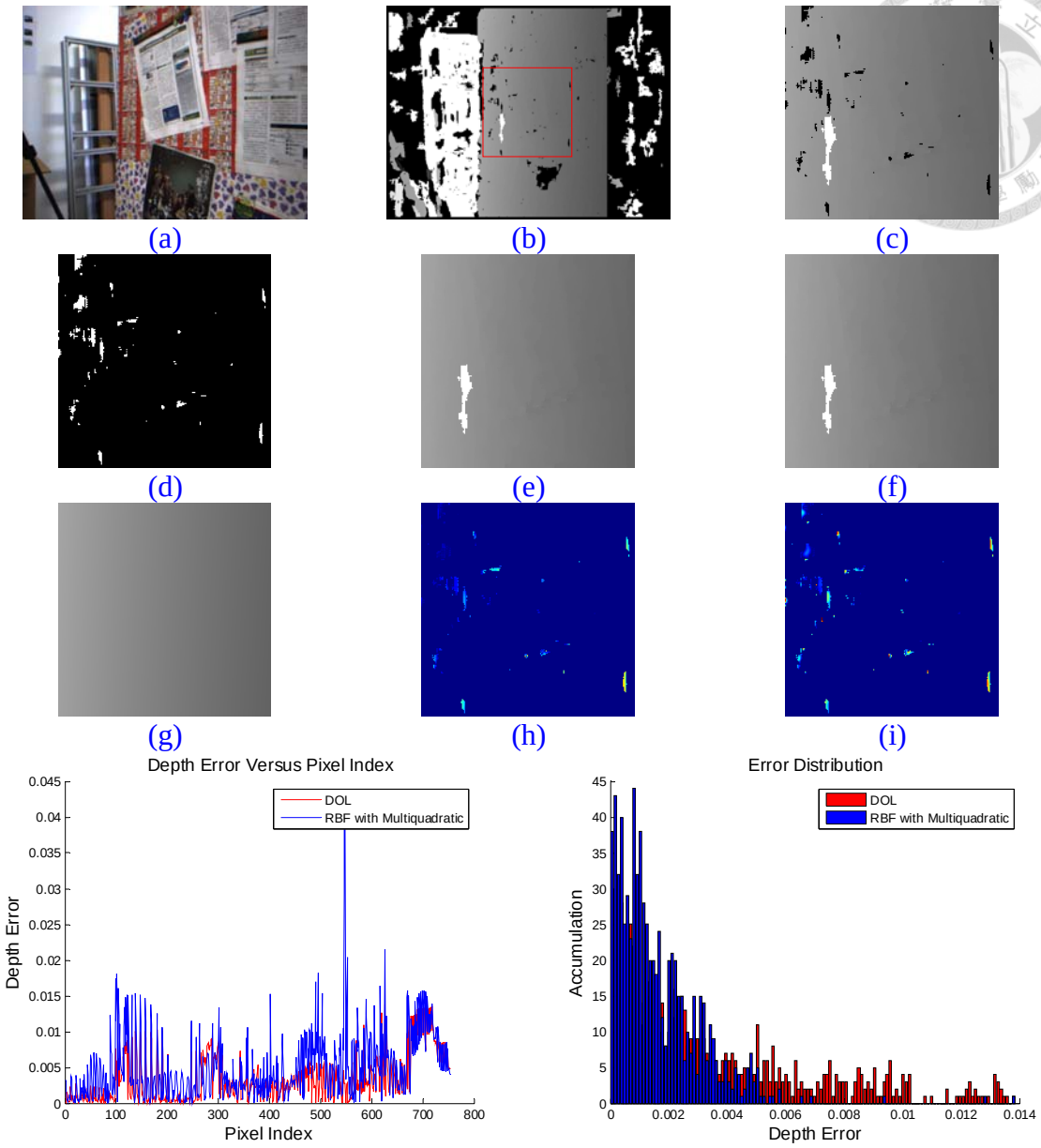


Figure 6.30: The data #1 interpolation result of two different approaches.

(a)-(c) The target image with corresponding raw depth map and the small patch extracted from the region enclosed by red rectangle in (b) to be analyzed.

(d) The white regions indicate the missing data area that will be interpolated.

(e) The interpolation result of the proposed DOL interpolation method.

(f) The interpolation result of the RBF interpolation method.

(g) The depth map estimated by using plane fitting method with the lasers data.

(h) The absolute interpolation error of DOL comparing to (g).

(i) The absolute interpolation error of RBF comparing to (g).

(j) Pixel index VS absolute depth error.

(k) The statistic result of the depth error, depth error VS number of pixels.

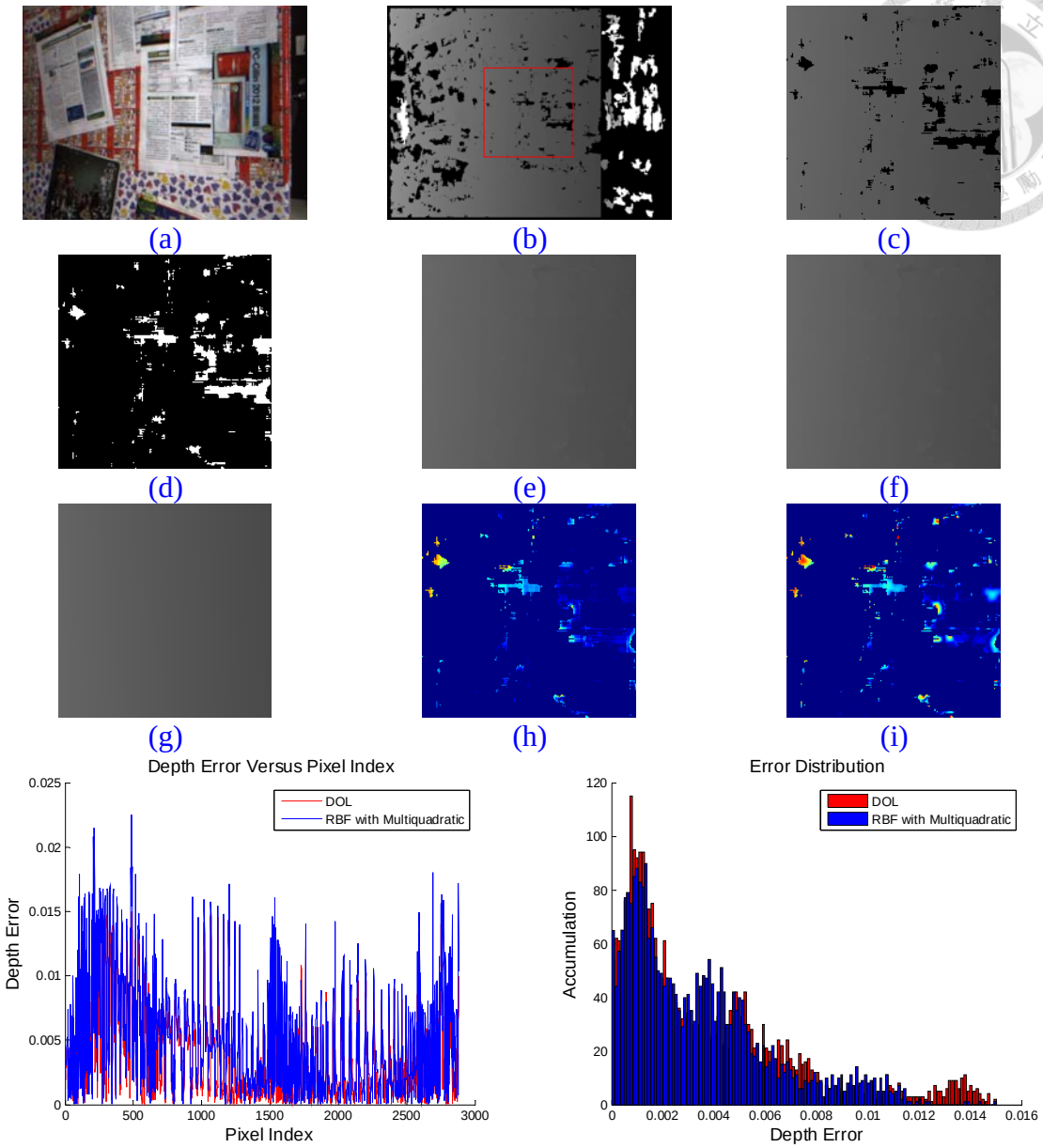


Figure 6.31: The data #2 interpolation result of two different approaches.

(a)-(c) The target image with corresponding raw depth map and the small patch extracted from the region enclosed by red rectangle in (b) to be analyzed.

(d) The white regions indicate the missing data area that will be interpolated.

(e) The interpolation result of the proposed DOL interpolation method.

(f) The interpolation result of the RBF interpolation method.

(g) The depth map estimated by using plane fitting method with the lasers data.

(h) The absolute interpolation error of DOL comparing to (g).

(i) The absolute interpolation error of RBF comparing to (g).

(j) Pixel index VS absolute depth error.

(k) The statistic result of the depth error, depth error VS number of pixels.

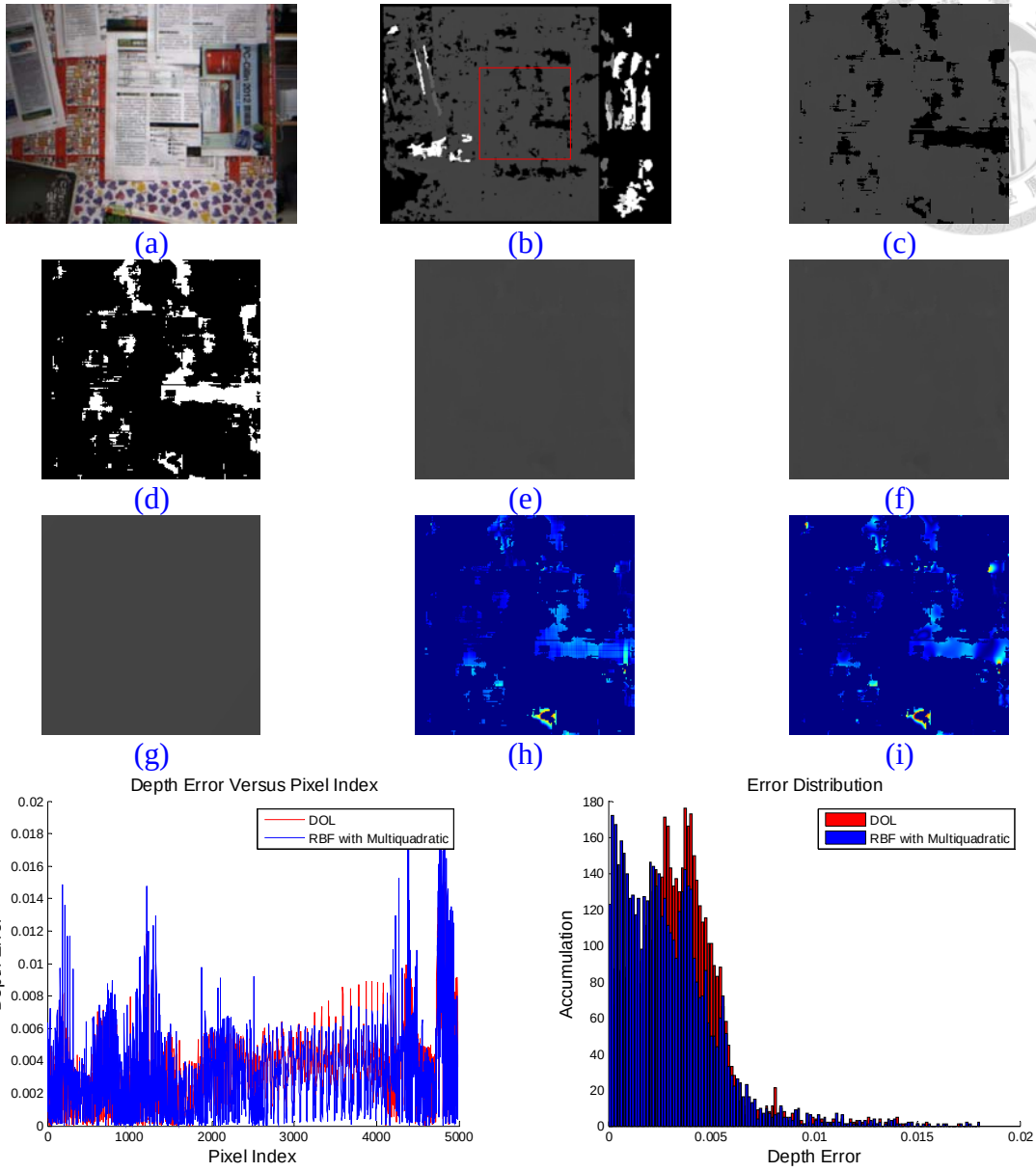


Figure 6.32: The data #3 interpolation result of two different approaches.

(a)-(c) The target image with corresponding raw depth map and the small patch extracted from the region enclosed by red rectangle in (b) to be analyzed.

(d) The white regions indicate the missing data area that will be interpolated.

(e) The interpolation result of the proposed DOL interpolation method.

(f) The interpolation result of the RBF interpolation method.

(g) The depth map estimated by using plane fitting method with the lasers data.

(h) The absolute interpolation error of DOL comparing to (g).

(i) The absolute interpolation error of RBF comparing to (g).

(j) Pixel index VS absolute depth error.

(k) The statistic result of the depth error, depth error VS number of pixels.

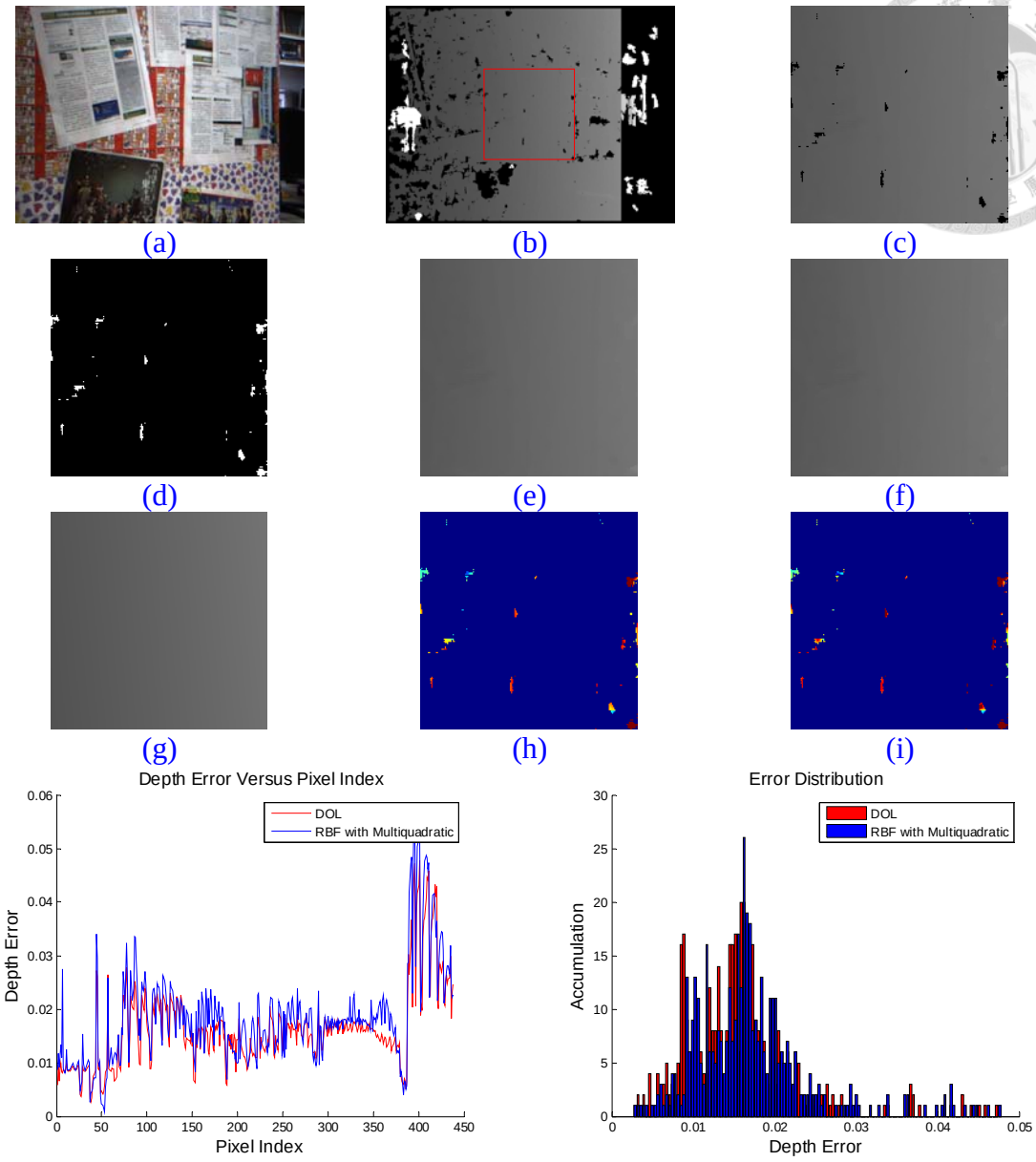


Figure 6.33: The data #4 interpolation result of two different approaches.

(a)-(c) The target image with corresponding raw depth map and the small patch extracted from the region enclosed by red rectangle in (b) to be analyzed.

(d) The white regions indicate the missing data area that will be interpolated.

(e) The interpolation result of the proposed DOL interpolation method.

(f) The interpolation result of the RBF interpolation method.

(g) The depth map estimated by using plane fitting method with the lasers data.

(h) The absolute interpolation error of DOL comparing to (g).

(i) The absolute interpolation error of RBF comparing to (g).

(j) Pixel index VS absolute depth error.

(k) The statistic result of the depth error, depth error VS number of pixels.

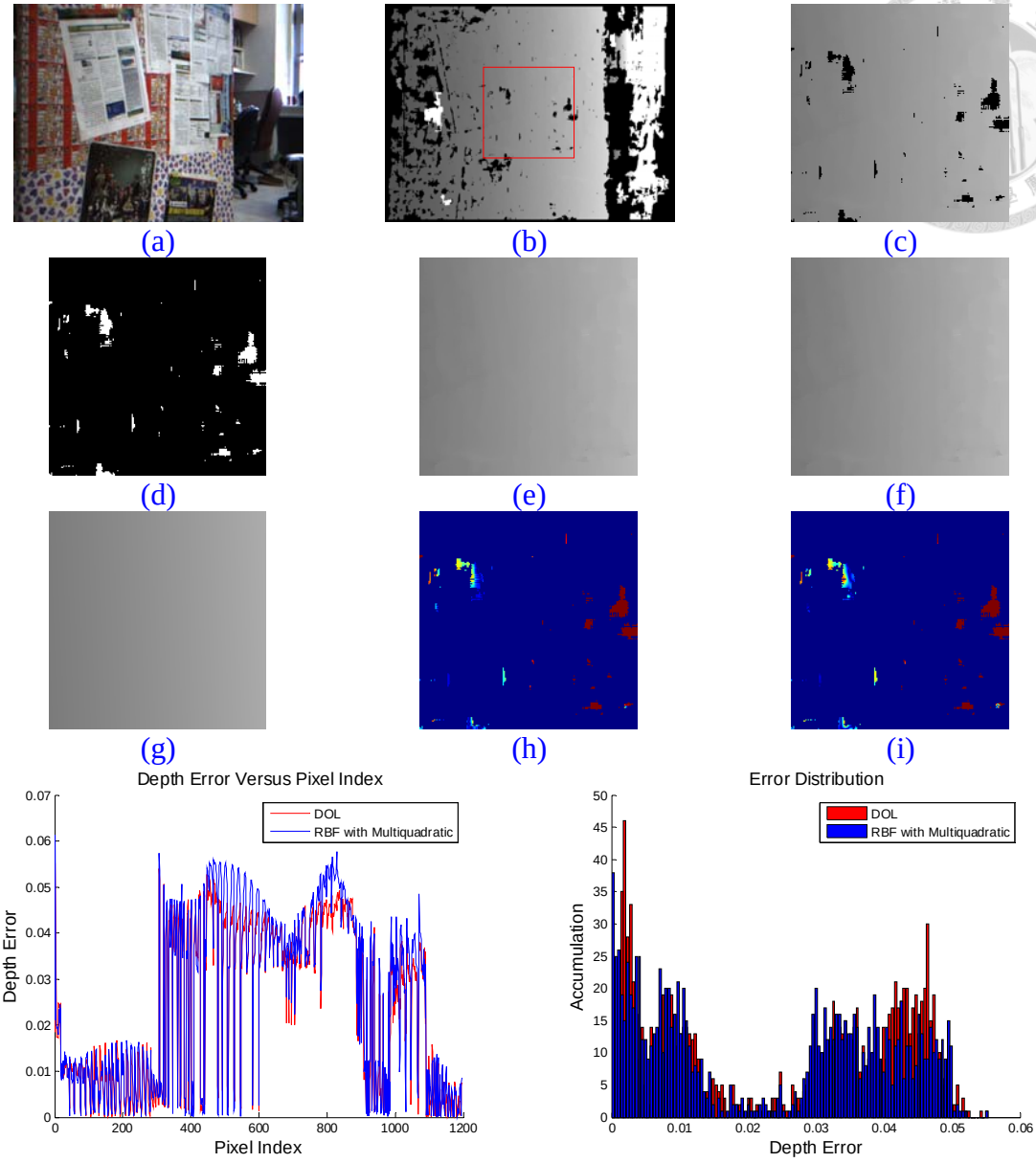


Figure 6.34: The data #5 interpolation result of two different approaches.

(a)-(c) The target image with corresponding raw depth map and the small patch extracted from the region enclosed by red rectangle in (b) to be analyzed.

(d) The white regions indicate the missing data area that will be interpolated.

(e) The interpolation result of the proposed DOL interpolation method.

(f) The interpolation result of the RBF interpolation method.

(g) The depth map estimated by using plane fitting method with the lasers data.

(h) The absolute interpolation error of DOL comparing to (g).

(i) The absolute interpolation error of RBF comparing to (g).

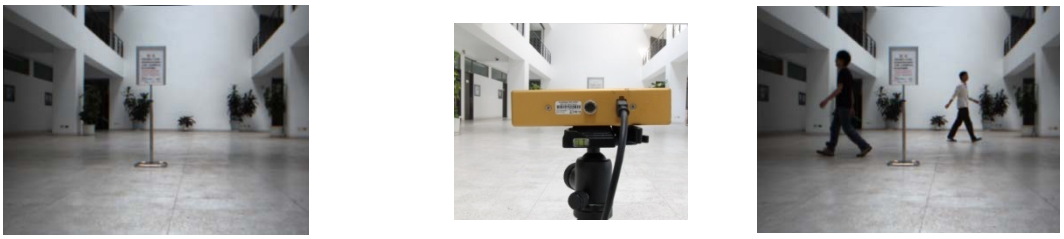
(j) Pixel index VS absolute depth error.

(k) The statistic result of the depth error, depth error VS number of pixels.

6.3 Object Detection and Tracking

In this section, the experimental result of the proposed object detection and tracking methods presented in Chapter 5 is shown and to be discussed. Subsection 6.3.1 shows the detection performance of the proposed system. Object tracking result is shown in Subsection 6.3.2. Two preset experimental scenarios are constructed to show the detection and tracking performance and accuracy.

The first preset experimental scenario is constructed at fifth floor in Ming-Da Hall as shown in Figure 6.35(a). Data are acquired by stereo camera with two people walking arbitrarily, as shown in Figure 6.35(c). The purpose in this experiment is to show the successful detection rate and tracking result with and without Kalman filter.



(a) (b) (c)

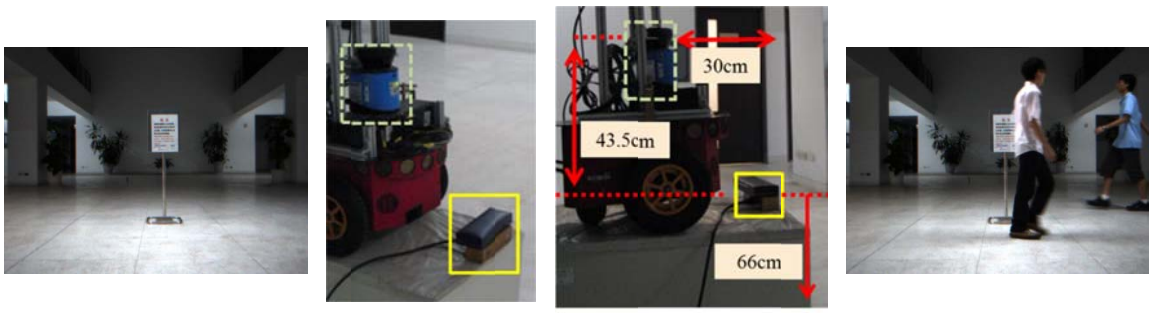
Figure 6.35: Experimental scenario setup

(a) Ming-Da Hall 5F

(b) Data are acquired from stereo camera in this experimental scenario.

(c) Two people are walking in the scenario.

The second experimental scenario is constructed at fifth floor in Ming-Da Hall which is the same as first experimental scenario, with a SICK laser scanner to measure the objects as benchmark. Figure 6.36 (b) and (c) shows the geometry relation between SICK laser scanner and stereo camera. Data are acquired by stereo camera and laser scanner with two people walking arbitrarily in front of laser scanner and stereo camera. The purpose in this experiment is to show the tracking accuracy of the proposed system.



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

Figure 6.36: Experimental scenario setup

(a) Ming-Da Hall 5F.

(b) Data are acquired from stereo camera (enclosed by solid line colored in light yellow) in this experimental scenario with SICK laser scanner (enclosed by dot line colored in light green) as benchmark.

(d) SICK laser scanner is mounted behind the stereo camera 30cm, and higher than stereo camera 4.35cm.

(c) Two people are walking in the scenario.

6.3.1 Object Detection

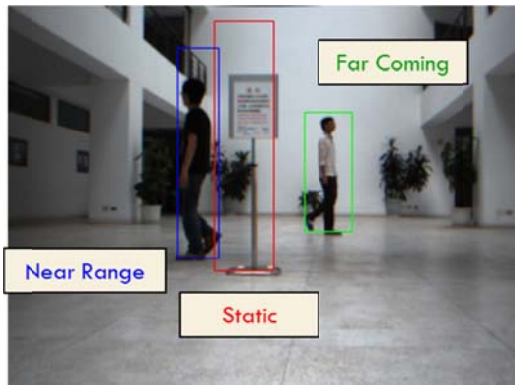
For total 81 frame data, successful detection rate of each object is calculated by the ratio between the number of correct detection result and the number of occurring in image in each frame manually. Three objects are defined as “Near Range”, “Far Coming” and ”Static” objects as illustrated in Figure 6.37(a), and the corresponding detection rates are shown in Figure 6.37(b). Note that in this analysis, only detection results are considered, tracking results are not discussed in this subsection.

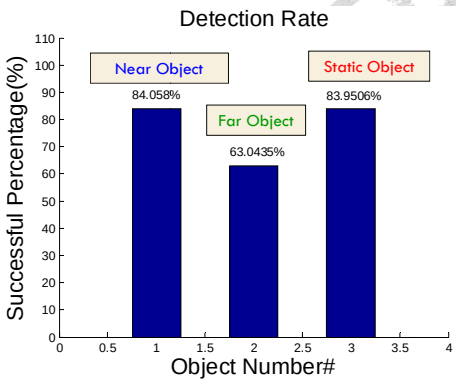
For near range object which is enclosed with blue bounding box, the successful detection rate is 84.058%. Two cases cause the false detection results are illustrated in Figure 6.39 and Figure 6.40. Figure 6.39 shows the first case that an object does not be detected since the lack of measurement pixels on to the object when it moves into the camera field of view. Figure 6.39(a)-(b) show the color images captured in time step 1 to 2, whereas Figure 6.39(c)-(d) is the corresponding disparity maps. Since the lack of disparity pixels on the object as shown in Figure 6.39(c), the width of the detecting object candidate is too short that is removed by the constraint shown in Equation (5.19), as shown in Figure 6.39(i) and (k). Another case that an object moves behind a static object is illustrated in Figure 6.40. Since the object moves too close to the stand, its disparity pixels are projected into the same u-disparity cell at where the stand is located. Thus the object has no corresponding measurement in current step as shown in Figure

6.40(l). For the static object which is enclosed with red bounding box, the successful detection rate is 83.95%, which is only affected by the occlusion case.

On the other hand, for the far coming object which is enclosed with green bounding box, the successful detection rate is only 63.0435%. The reason is that not only the object is affected by the cases of entering the camera FOV and occlusion like the near range object, but it has additional case that the object cannot be detected since the disparity constraint mentioned in Subsection 5.2.2. Although the object is in the image plane, it is not considered as a valid candidate.

The proposed object detection method is also compared to the existing human detection method based on Histogram-of-Oriented Gradients (HOG) with SVM classifier proposed in [21: Dalal et al. 2005] and the source code is found in the open source library called OpenCV in version 2.4.3 [57: OpenCV from OpenCV official website 2013]. Because the HOG human detection is a training based method and focuses on detecting human object, the static stand object cannot be detected. For near object which is a human dressed in black, the detection rate is 72.46%, which is lower than the proposed object detection method. The reason is that the average gradient image over the training examples is a frontal-like human gradient image. The failure occurs when human is seen from lateral side or when the human is walking causes the shape is not similar to the gradient image due to the human arms are waving.





(a) (b)

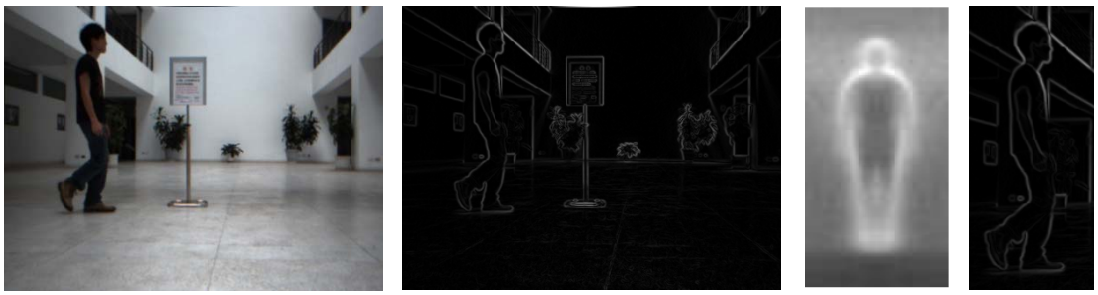
Figure 6.37: Detection rate of each object.

(a) The object definition.

(b) The detection rate of each object defined by (a).

Table 6.7: Successful detection rate of each object.

Object	Near Range	Far Coming	Static
The Proposed Method Detection Rate (# Detection / #In Image Plane)	58/69 (84.06%)	29/46 (63.04%)	68/81 (83.95%)
Histogram of Gradient (HOG) Detection Rate (# Detection / #In Image Plane)	50/69 (72.46%)	22/46 (47.82%)	None



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

Figure 6.38: An example of histogram-of-oriented gradients (HOG) method failure detection case.

(a)(b) Testing image and its magnitude of gradient image.

(c) The average gradient image over the training examples in [21: Dalal et al. 2005].

(d) The object gradient patch image. It is not similar to the average gradient image in (c).

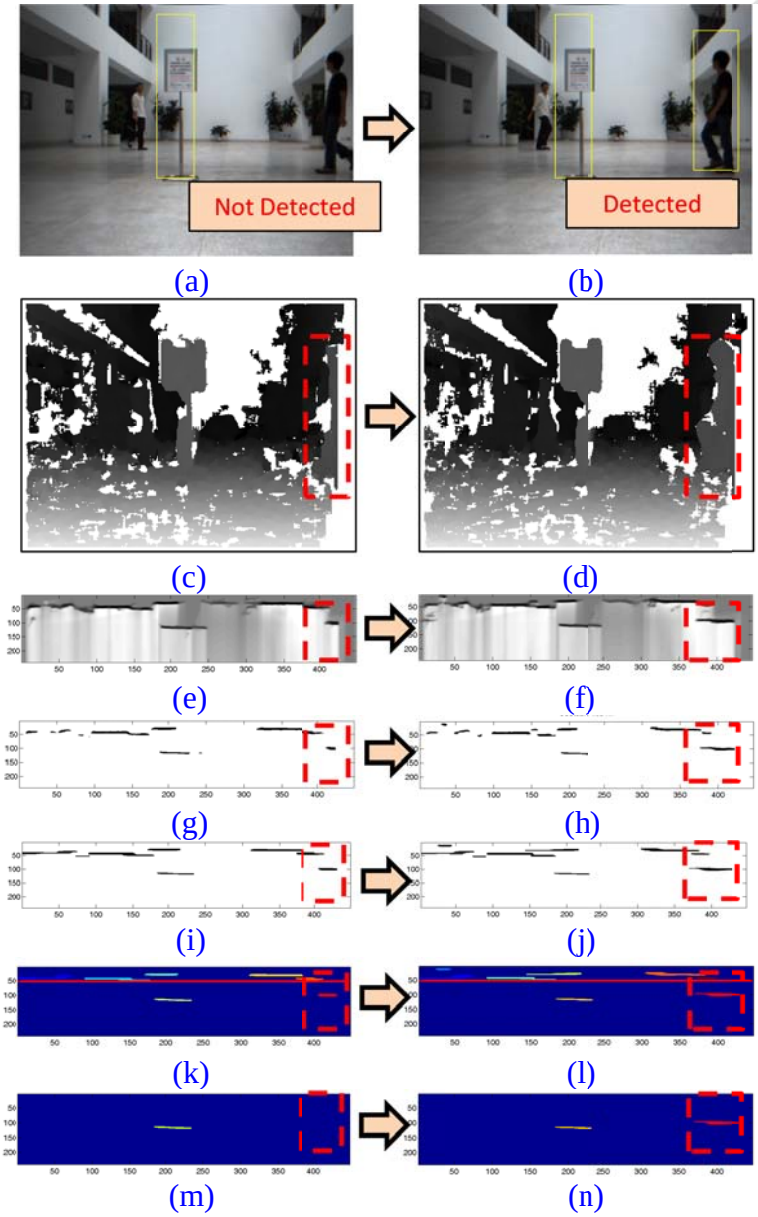


Figure 6.39: For the near range object, one of two cases that is considered to be false detection for example.

(a)(b) When an object moves into the camera field of view, the proposed method cannot detects the object successfully since the candidate width is too short and is eliminated by the width constraint as show in (m).

(c)(d) The corresponding disparity map.

(e)(f) The occupancy grid with Gaussian filtering.

(g)(h) The binary grid extracted by using the probability threshold.

(i)(j) The result of morphological process at (g) and (h).

(k)(l) The result of connected-component labeling.

(m)(n) Eliminate the candidate whose disparity is smaller than $d_{\min}$ and the width is smaller than W_{Th}.

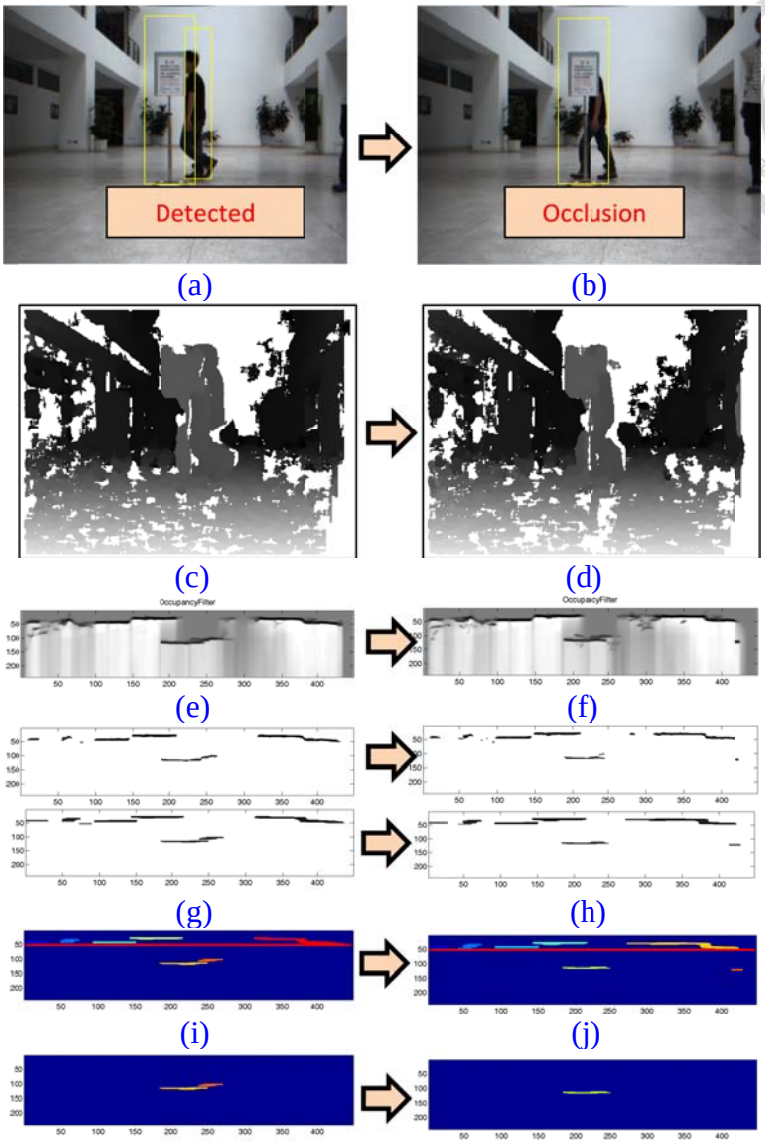


Figure 6.40: For the near range object, one of two cases that is considered to be false detection for example.

(a)(b) When an object moves back to an object, the bounding box cannot enclose the object successfully.

(c)(d) The corresponding disparity map.

(e)(f) The occupancy grid with Gaussian filtering.

(g)(h) The binary grid extracted by using the probability threshold.

(i)(j) The result of morphological process at (g) and (h).

(k)(l) The result of connected-component labeling.

(m)(n) Eliminate the candidate whose disparity is smaller than $d_{\min}$ and the width is smaller than W_{Th}.

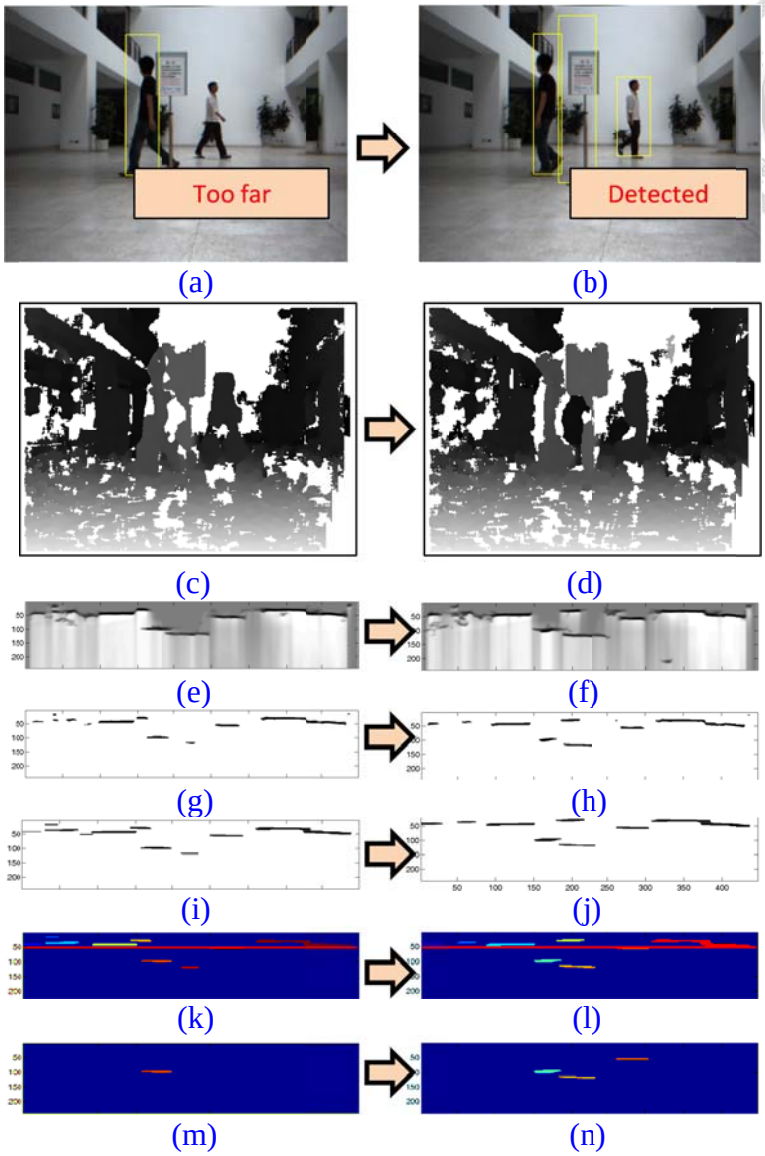


Figure 6.41: The far coming object not only has the false detection cases of entering the image plane and the occlusion, it also has the case that is too far to be detected.

(a)(b) The object moves close to the camera. Since the preset disparity threshold, the object cannot be detected in the previous step shown in (a), and illustrated in (m).

(c)(d) The corresponding disparity map.

(e)(f) The occupancy grid with Gaussian filtering.

(g)(h) The binary grid extracted by using the probability threshold.

(i)(j) The result of morphological process at (g) and (h).

(k)(l) The result of connected-component labeling.

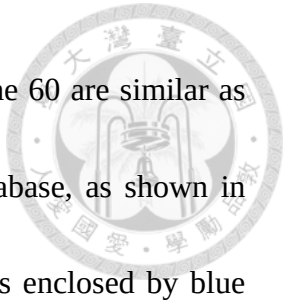
(m)(n) Eliminate the candidate whose disparity is smaller than $d_{\min}$ and the width is smaller than W_{Th}.

6.3.2 Object Tracking

The detection rate is discussed in Subsection 6.3.1. Although object is not detected in every step, this subsection will show that an object can be tracked by the proposed method without losing it even when the object returns to the camera FOV or is occluded temporarily. Moreover, the benefit of using Kalman filter in the tracking task is also discussed in this subsection.

First of all, Figure 6.42 shows the tracking result in the image plane and the X-Z Cartesian space with corresponding color. For simplicity, only 14 tracking results are shown at four frame intervals (except the frame 32 to 38). It can be observed that each object is tracked correctly in the image plane and the corresponding position in the X-Z coordinate. Frame 24-38 show that even though the object which is enclosed by blue bounding box moves out of the camera field of view, it can be tracked successfully when it returns to the camera FOV. On the other hand, frame 58-66 show that even though the object moves behind the stand and is occluded, it can be tracked successfully when it moves out of stand and is measured by the camera again.

The overall tracking results on the X-Z plane with and without Kalman filter are shown in Figure 6.43 and Figure 6.44. It can be observed that a sparkle is occurred in the Figure 6.44. This is because two objects are too close to each other and are considered to be the same candidate in frame 60, as shown in Figure 6.45. The



distributions of hue and saturation channels of the candidate of frame 60 are similar as the distributions of the human who wears a black cloth in the database, as shown in Figure 6.46. Therefore, the candidate is linked to the human who is enclosed by blue bounding box as shown in Figure 6.46(g). Since the coordinate of each point on the stand contribute to the object, the coordinate of the human is dragged close to the coordinate of the stand. Applying Kalman filter can eliminate this problem since it is not only considering the measurement but also the motion model. The tracking results of each object are shown in Figure 6.47, Figure 6.48 and Figure 6.49. From these figures it can be observed that Kalman filter do not improve the result dramatically. Note that the x-coordinate of object at frame 60 shown in Figure 6.48, this is sparkle signal which is eliminated by applying Kalman filter.

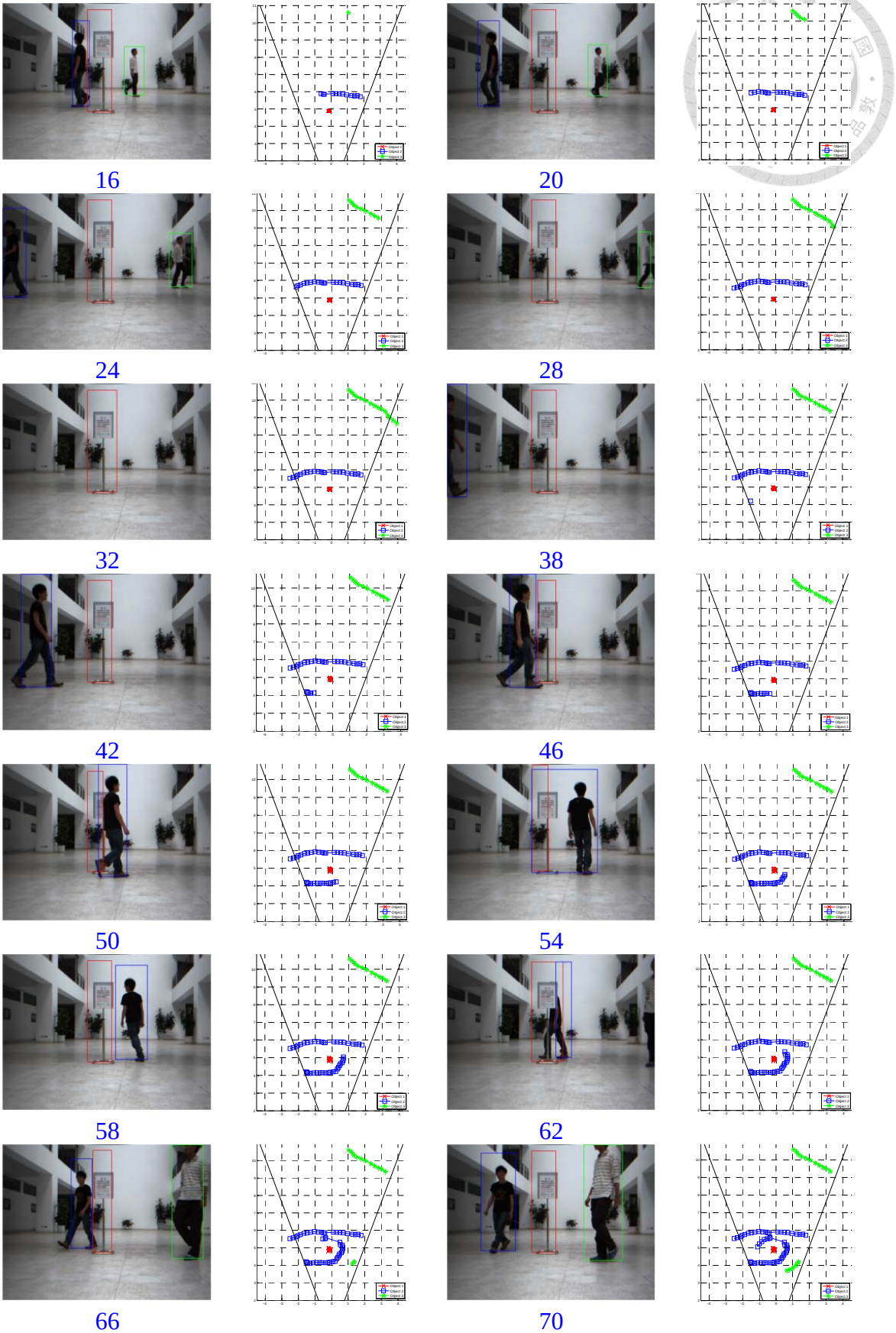


Figure 6.42: Tracking result in image space and Cartesian space

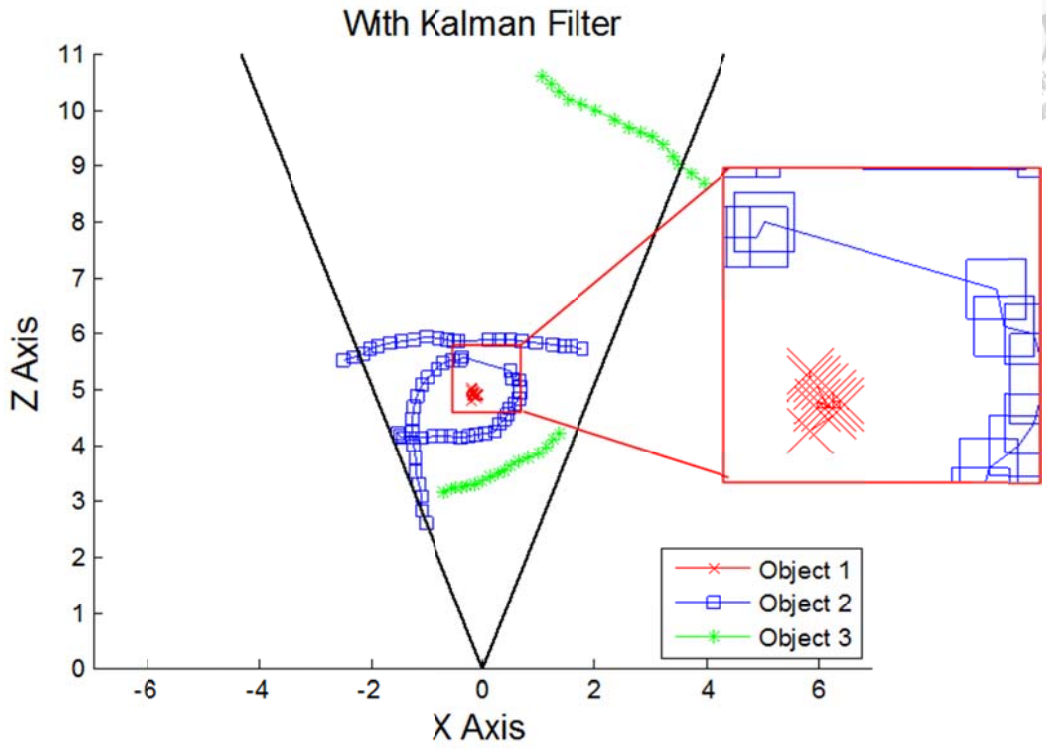


Figure 6.43: Object tracking result with applying Kalman filter

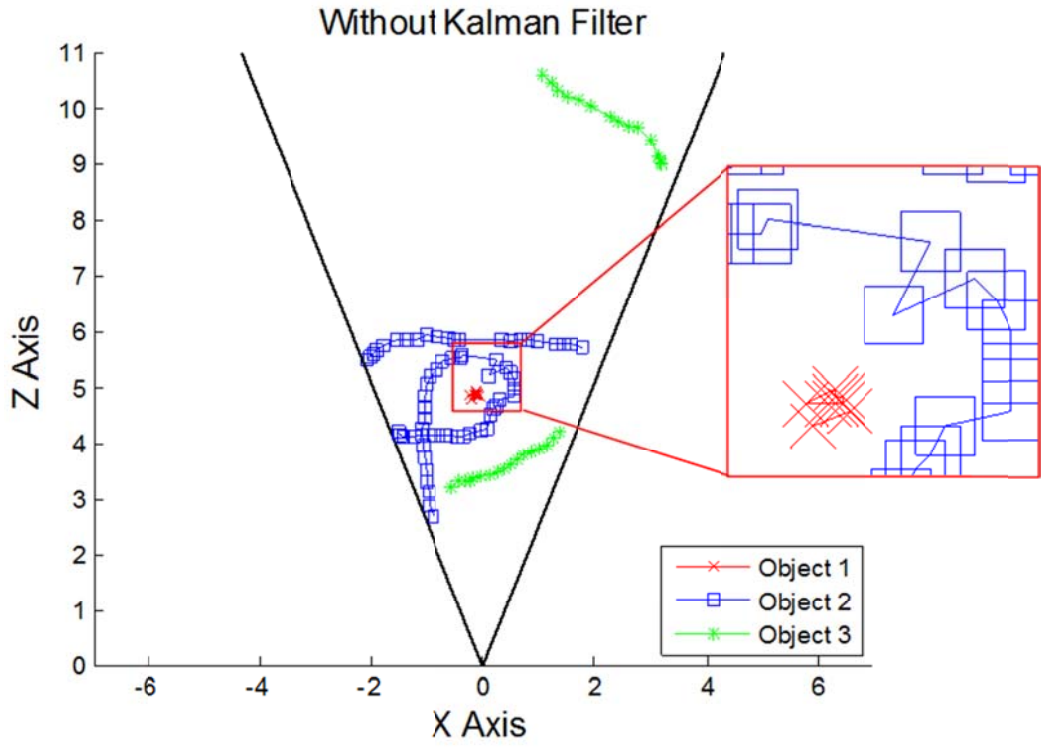
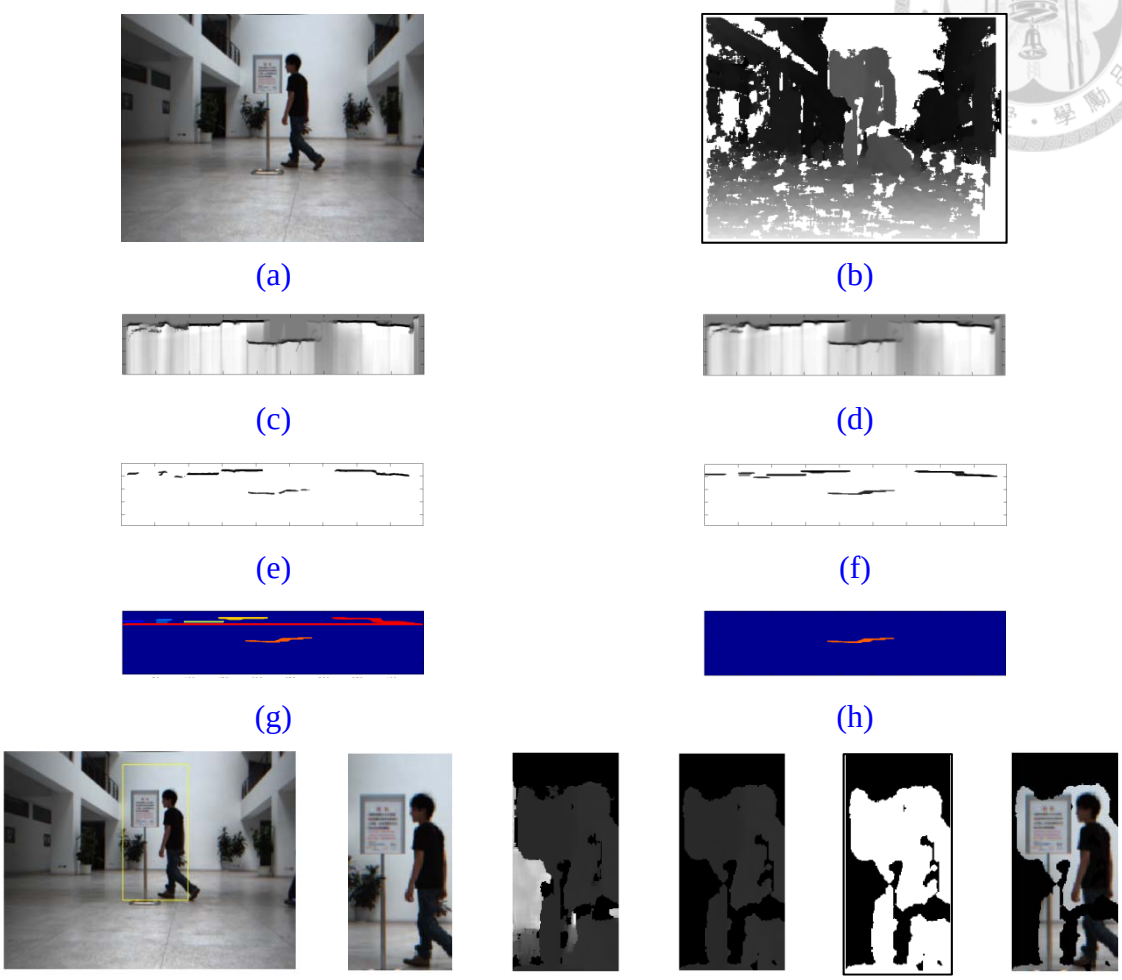


Figure 6.44: Object tracking result without applying Kalman filter



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j) (k) (l) (m) (n)

Figure 6.45: Too close objects are measured as the same candidate in frame 60.

(a)(b) The target image and the disparity map.

(c)-(h) Each step of post-processing mentioned in Subsection 5.2.2.

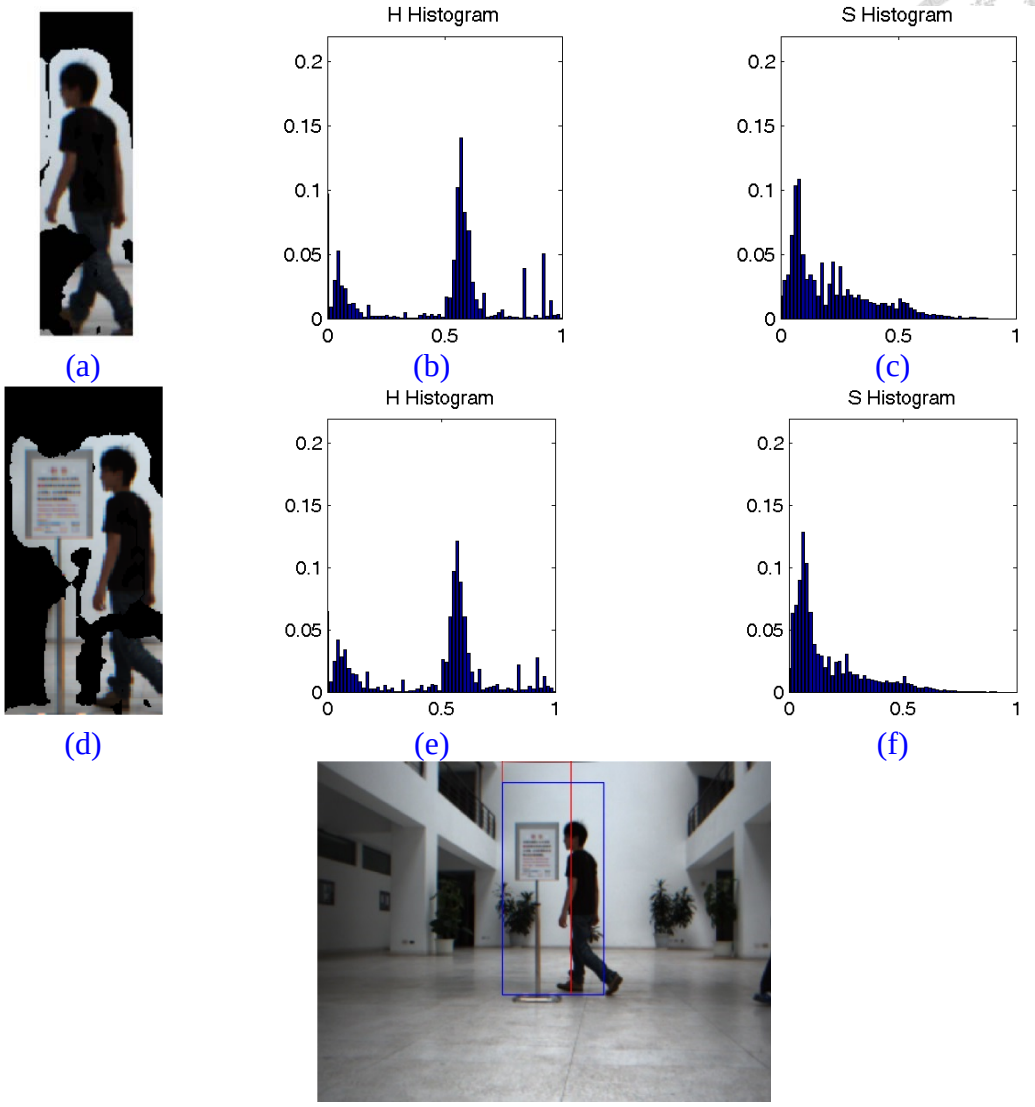
(i) The candidate enclosed by its bounding box. Two objects are enclosed by the same bounding box.

(j)(k) The image patch and the depth map of the candidate.

(l) The depth map with background elimination.

(m) The remaining foreground mask.

(n) The image patch with background elimination by (m). Note that in this case two objects are too close that cannot be separated from each other.



(a) (b) (c)

(d) (e) (f)

(g)

Figure 6.46: The candidate in frame #59 and #60 and the data association result.

(a) The residual image of frame image #59 after the background removing step.

(b)(c) The hue and saturation distributions of (a).

(d) The residual image of frame image #60 after the background removing step.

(e)(f) The hue and saturation distributions of (d).

(g) Since the hue and saturation distributions of the candidate in frame #60 are close to the hue and saturation distributions of the human object in the database, data association mechanism proposed in this thesis may fail in this case.

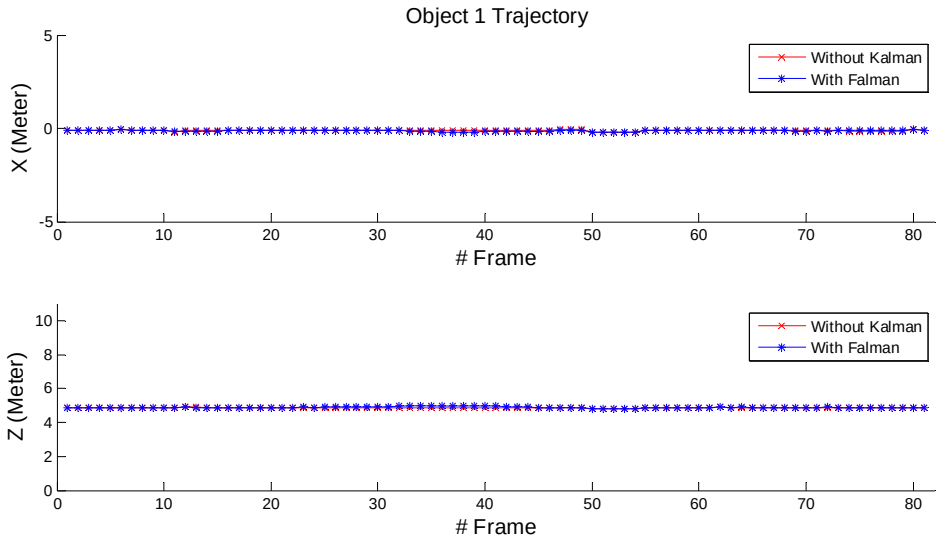


Figure 6.47: Position of object #1 with and without Kalman Filter

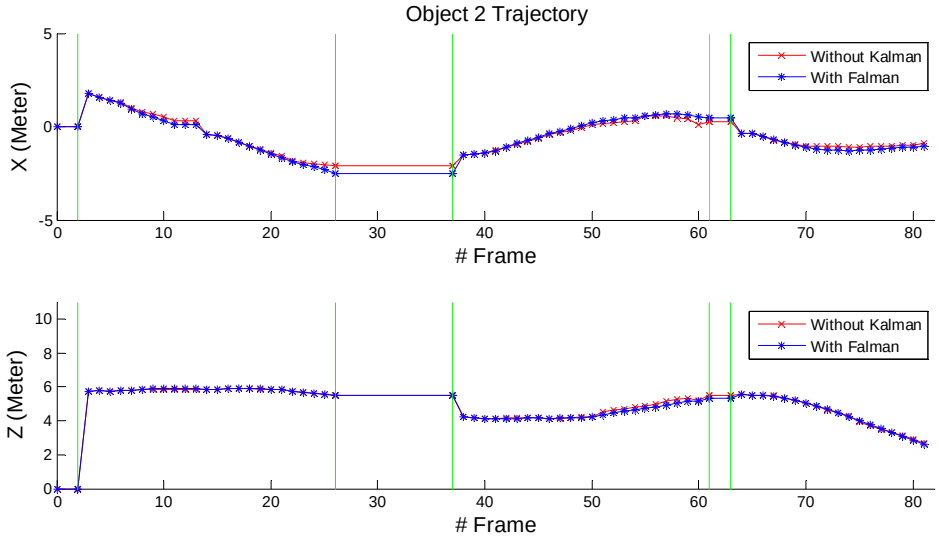


Figure 6.48: Position of object #2 with and without Kalman Filter

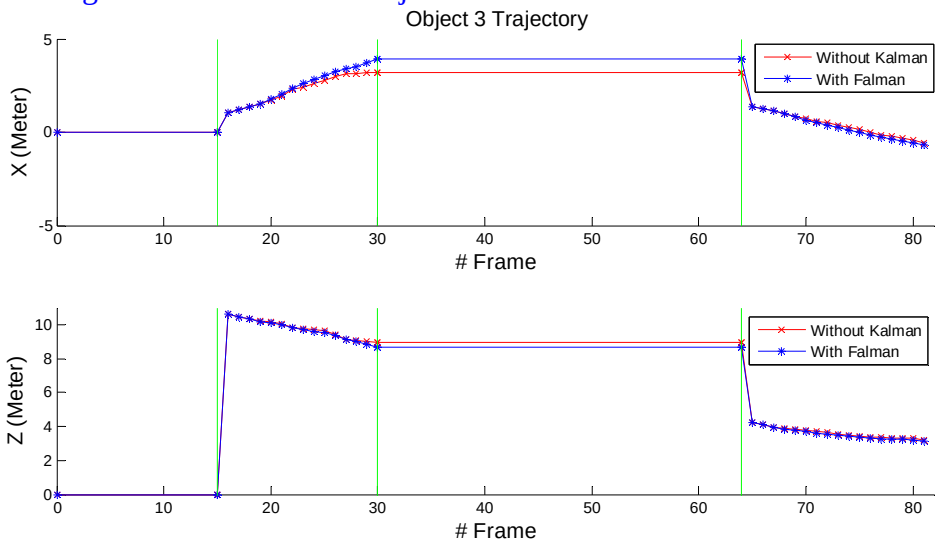
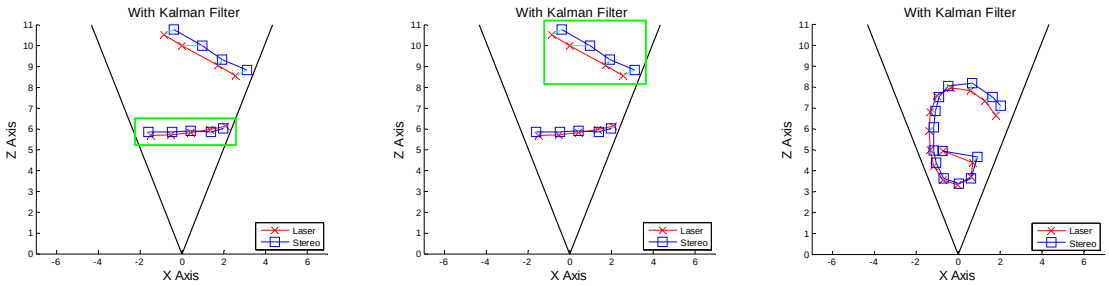


Figure 6.49: Position of object #3 with and without Kalman Filter

Tracking Accuracy

In this experiment, the accuracy of the proposed system is compared to SICK laser scanner. The object path is decomposed to three parts to analyze the accuracy in different conditions, which are “near path,” “far path” and “circular path,” as shown in Figure 6.50. The tracking results of each part are listed in Table 6.8, Table 6.9 and Table 6.10. For the near path object, the mean of object to laser distances is 0.1468 and the standard deviation is 0.0457. It is acceptable distance for a non-rigid object since the center of the object may change when it moves, and the laser measurement on it may hit at the different part of the object body. For the far path, the mean of object to laser distances is 0.6140 and the standard deviation is 0.2639. It is obvious that the mean and standard deviation of the far object are larger than the near range object. This is reasonable since the stereo uncertainty increases when the distance of the measuring object is getting larger. For the circular path, the mean of object to laser distances is 0.2219 and the standard deviation is 0.1444. Its mean and standard deviation are increased comparing to the near path object. This is because that applying Kalman filter with constant velocity motion model, the velocity of an object is assumed to be a constant. However, an object moves in a circular path is not a constant velocity since the direction of the motion is changing all the time. Therefore, when an object moves along a circular-like shape path, the proposed tracking method applying Kalman filter with

constant velocity model is not inaccurate. It may get the better result if the motion model is replaced by other nonlinear model and will be the system future work.



(a) (b) (c)

Figure 6.50: Object trajectories estimated by the proposed method and measured by SICK LMS100 laser scanner. The blue square signs indicate the object positions in each step estimated by the proposed tracking method using stereo camera, whereas the red cross signs represent the object positions in each step measured by the SICK LMS100 laser scanner. The path is divided into three parts:

(a) The near path.
(b) The far path.
(c) The circular path.

Table 6.8: The tracking result of near path (m)

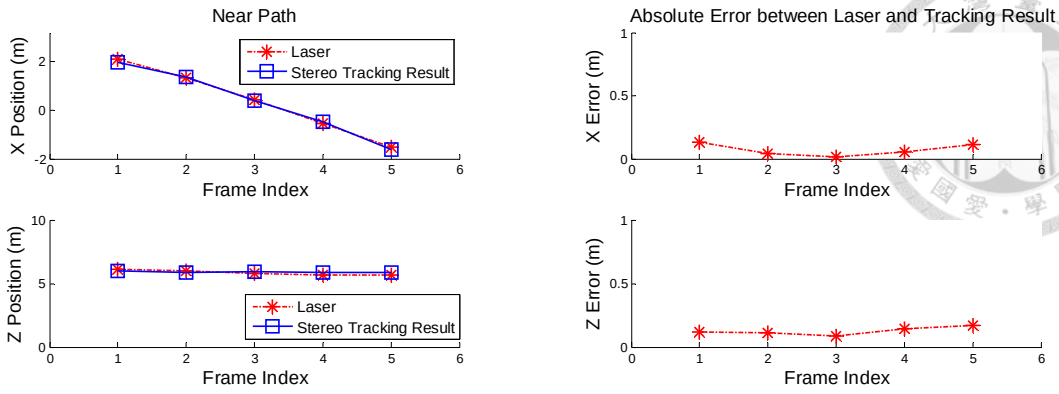
	Laser		Stereo with proposed method		Error		
Step / Axis	X Axis	Z Axis	X Axis	Z Axis	X Axis	Z Axis	Distance
Step 1	2.095	6.118	1.965	6.001	0.1300	0.1170	0.1749
Step 2	1.338	5.965	1.381	5.856	0.0430	0.1090	0.1172
Step 3	0.4298	5.817	0.412	5.901	0.0178	0.0840	0.0859
Step 4	-0.5279	5.704	-0.4705	5.848	0.0574	0.1440	0.1550
Step 5	-1.497	5.674	-1.607	5.842	0.1100	0.1680	0.2008
Mean	N/A	N/A	N/A	N/A	0.0716	0.1244	0.1468
STD	N/A	N/A	N/A	N/A	0.0469	0.0324	0.0457

Table 6.9: The tracking result of far path (m)

	Laser		Stereo with proposed method		Error		
Step / Axis	X Axis	Z Axis	X Axis	Z Axis	X Axis	Z Axis	Distance
Step 1	2.543	8.538	3.104	8.809	0.5610	0.2710	0.6230
Step 2	1.735	9.034	1.911	9.322	0.1760	0.2880	0.3375
Step 3	6.31E-16	9.975	0.9674	9.989	0.9674	0.0140	0.9675
Step 4	-0.8536	10.52	-0.3888	10.77	0.4648	0.2500	0.5278
Mean	N/A	N/A	N/A	N/A	0.5423	0.2057	0.6140
STD	N/A	N/A	N/A	N/A	0.3272	0.1288	0.2639

Table 6.10: The tracking result of circular path (m)

	Laser		Stereo with proposed method		Error		
Step / Axis	X Axis	Z Axis	X Axis	Z Axis	X Axis	Z Axis	Distance
Step 1	1.800	6.632	2.026	7.111	0.2260	0.4790	0.5296
Step 2	1.282	7.332	1.629	7.512	0.3470	0.1800	0.3909
Step 3	0.570	7.821	0.660	8.173	0.0900	0.3520	0.3633
Step 4	-0.362	7.965	-0.485	8.028	0.1230	0.0630	0.1382
Step 5	-1.039	7.559	-0.903	7.521	0.1360	0.0380	0.1412
Step 6	-1.322	6.805	-1.076	6.847	0.2460	0.0420	0.2496
Step 7	-1.379	5.892	-1.177	6.081	0.2020	0.1890	0.2766
Step 8	-1.328	4.993	-1.157	4.961	0.1710	0.0320	0.1740
Step 9	-1.175	4.214	-1.049	4.386	0.1260	0.1720	0.2132
Step 10	-0.717	3.540	-0.699	3.616	0.0180	0.0760	0.0781
Step 11	-0.032	3.310	0.034	3.365	0.0660	0.0550	0.0859
Step 12	0.602	3.696	0.622	3.615	0.0200	0.0810	0.0834
Step 13	0.703	4.374	0.922	4.642	0.2190	0.2680	0.3461
Step 14	-0.712	4.936	-0.748	4.937	0.0360	0.0010	0.0360
Mean	N/A	N/A	N/A	N/A	0.1447	0.1449	0.2219
STD	N/A	N/A	N/A	N/A	0.0964	0.1394	0.1444

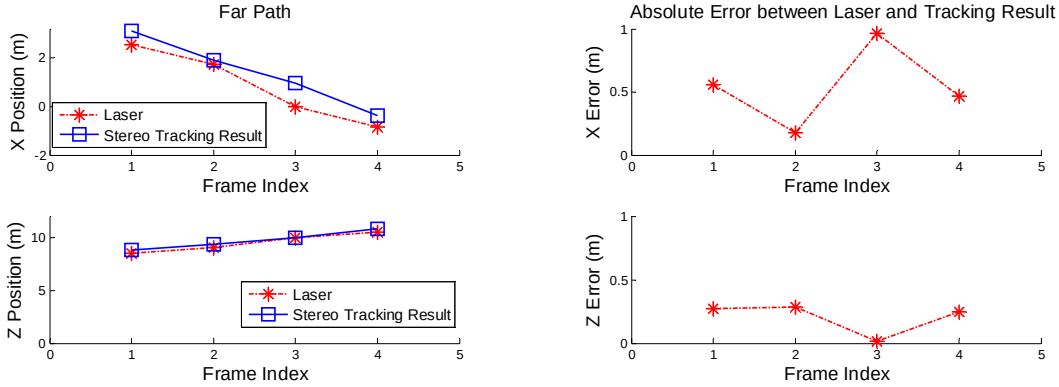


(a) (b)

Figure 6.51: Near path object tracking result comparing to laser.

(a) The position of each step.

(b) The absolute error of each step comparing to laser.

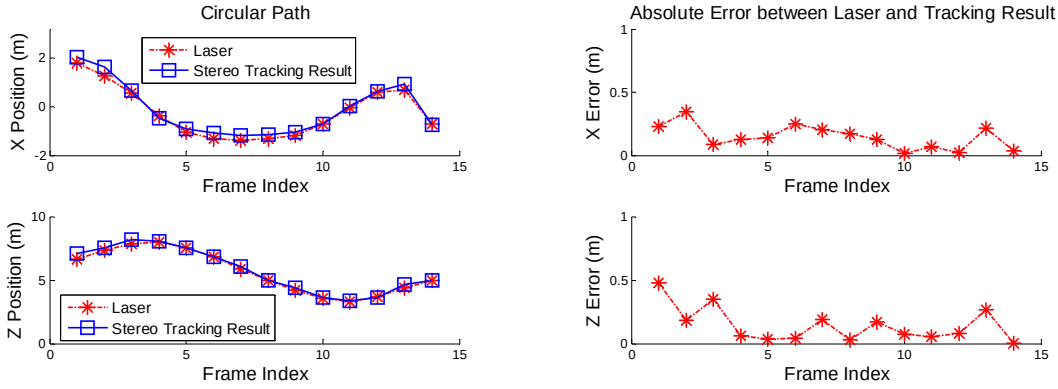


(a) (b)

Figure 6.52: Far path object tracking result comparing to laser.

(a) The position of each step.

(b) The absolute error of each step comparing to laser.



(a) (b)

Figure 6.53: Circular path object tracking result comparing to laser.

(a) The position of each step.

(b) The absolute error of each step comparing to laser.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

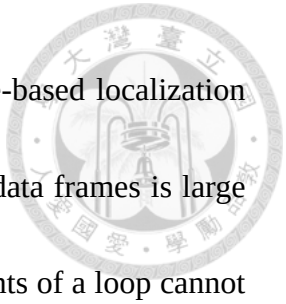
In this thesis, the feature-based RGB-D localization method is presented to localize stereo camera. This method uses image features to connect the relationship between two consecutive frame data. Thus the camera relative pose can be estimated by using SVD decomposition method with matching feature pairs with RANSAC outlier rejection. The experiment shows that the localization method is quite robust by using two orthogonal laser range finders as a benchmark, and the localization result is suitable in the mapping task whereas the mapping quality is evaluated in visual aspect. Moreover, the proposed stereo refinement method eliminates the wrong pixels in the occlusion area, and the small missing data area which is called a hole is interpolated by two different methods, which are the proposed dual orthogonal linear (DOL) and radial basis function interpolation. Experiment results show that by applying the refinement method, 3D model data is increased 2% in the image space, and the accuracies of two interpolation methods are in acceptable range in the planar case. Experiment results also show that the processing time of the proposed DOL interpolation method is three time faster than

RBF.

On the other hand, the proposed object detection and tracking system is proposed to track multiple objects. The visibility-based occupancy grid map construction method proposed in [29: Perrollaz et al. 2012] can estimate the probability of each grid cell in u-disparity space. The proposed data post-processing method eliminates the noise and the region of the candidate in the occupancy grid map can be extracted by using the connected-component labeling technique and then the bounding box used to enclose the corresponding object in image can be obtained. Objects in the database can be successfully registered to these candidates by comparing the distributions of hue and saturation channels as object feature vectors. Finally, each database object can be renewed by the proposed update strategy. Experiment results show that even though an object returns to the field of view or is in occlusion, the object can still be tracked correctly.

7.2 Future Work

Most complete mapping systems require three considerations, which are the spatial alignment of consecutive data frames to achieve localization task, the detection of loop closures and the globally consistent alignment of all data frames [1: Henry et al. 2012]. This thesis implements the 3D mapping system considering the spatial alignment



without the loop detection and global consistency. Since the feature-based localization method is processing frame-by-frame, the accumulating drift of all data frames is large when the camera moves for a long distance and therefore the endpoints of a loop cannot be aligned together. This is the main problems of the proposed 3D model reconstruction system of this thesis that should be solved in the future. Besides, since the geometry of binocular stereo camera is fixed, the physical relationship between left and right images can be another constraint to make the localization result more accurate as proposed in [18: Kitt et al. 2010]. Moreover, the proposed 3D model reconstruction system does not consider how to model and update the mapping data. In [1: Henry et al. 2012], each point in 3D model are transformed to the “surface element (Surfel)” data structure with the proposed update strategy. With Surfel mapping model and update strategy, not only the visualization result is improved by using surface representation, but also make the update task more easily.

On the other hand, for the proposed stereo refinement method, hole region to be filled is selected by a fix threshold currently. However, due to the properties of camera projection, the size of a certain hole changes according to the distance to the camera coordinate. Therefore, a dynamic range of the filling hole selection mechanism based on the measurement distance to the camera coordinate is the future work to improve the method.

For the proposed object detection and tracking system, three aspects can be improved and extended. First of all, this thesis use visibility-based occupancy grid to detect object. However, the advantage of Bayesian occupancy filter (BOF) framework does not implement currently in thesis. With BOF framework, a static global map can be constructed by several frame data, and then moving object can be filtered out by comparing the local u-disparity occupancy grid map to the global occupancy grid map. The same concept implemented in Cartesian space using laser range finder has been proposed in [34: Wolf et al. 2004]. Secondly, as the experiment results mentioned in Subsection 6.3.2, the Kalman filter with constant velocity motion model is not quite accurate when an object moves along a circular path. To overcome this problem, extended Kalman filter (EKF) with nonlinear dynamic model might be a solution. Third, the proposed object tracking system has not been integrated into the proposed 3D environment reconstruction system. Combining the localization method in the first topic mentioned in Section 4.1 and the u-disparity occupancy grid with BOF framework to handle the dynamic environment is the next work of this thesis.

References

[1: Henry et al. 2012]

Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, Dieter Fox, "RGB-D Mapping: Using Kinect-style Depth Cameras for Dense 3D Modeling of Indoor Environments," International Journal of Robotics Research, vol. 31, no. 5, pp. 647-663, April 2012.

[2: Marcincin et al. 2012]

J.Novak-Marcincin, J. Torok, J. Barna, M. Janak, L. Novakova-Marcincinova and V. Fecova, "Realization of 3D Models for Virtual Reality by Use of Advanced Scanning Methods," in Proceedings of IEEE International Conference on Cognitive Infocommunications, pp. 787-790, December 2-5, 2012.

[3: Park et al. 2012]

DongRyeol Park, Joon-Kee Cho and Yeon-Ho Kim, "A Visual Guidance System for Minimal Invasive Surgery Using 3D Ultrasonic and Stereo Endoscopic Images," in Proceedings of IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, Roma, Italy, pp. 872-877, June 24-27, 2012.

[4: Noonan et al. 2009]

David P. Noonan, Peter Mountney, Daniel S. Elson, Ara Darzi and Guang-Zhong

Yang, “A Stereoscopic Fibroscope for Camera Motion and 3D Depth Recovery during Minimally Invasive Surgery,” in Proceedings of IEEE Conference on Robotics and Automation, Kobe, Japan, pp. 4463-4468, May 12-17, 2009.

[5: Zeisl et al. 2012]

Bernhard Zeisl, Kevin Koser and Marc Pollefeys, “Viewpoint Invariant Matching via Developable Surfaces,” in Proceedings of the 12th International Conference on Computer Vision, pp. 62-71, 2012.

[6: Suarez et al. 2012]

Jesus Suarez and Robin R. Murphy, “Using the Kinect for Search and Rescue Robotics,” in Proceedings of the 2012 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), pp. 1-2, November 5-8, 2012.

[7: Hu et al. 2012]

Gibson Hu, Shoudong Huang, Liang Zhao, Alen Alepijevic and Gamimi Dissanayake, “A robust RGB-D SLAM algorithm,” in Proceedings of IEEE International Conference on Intelligent Robots and Systems, Vilamoura, pp. 1714-1719, 7-12 October, 2012.

[8: Murray et al. 2005]

Don Murray and James J. Little, “Patchlets: Representing Stereo Vision Data with Surface Elements,” in Proceedings of IEEE International Workshop on

Applications of Computer Vision, Breckenridge, CO, vol. 1, pp. 192-199, 5-7 January, 2005.

[9: Bsel et al. 1992]

Paul J. Bsel and Neil D. McKay, "A Method for Registration of 3D Shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no.2, pp. 239-256, February 1992.

[10: Turk et al. 1994]

Greg Turk and Marc Levoy, "Zippered Polygon Meshes from Range Images," in Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, New York, USA, pp. 311-318, July 24-29, 1994.

[11: Chen et al. 1991]

Yang Chen and Gerard Medioni, "Object Modeling by Registration of Multiple Range Images," in Proceedings of IEEE International Conference on Robotics and Automation, Sacramento, CA, vol. 3, pp. 2724-2729, April 9-11, 1991.

[12: Johnson et al. 1997]

Andrew Edie Johnson and Sing Bing Kang, "Registration and Integration of Textured 3D Data," in Proceedings of IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Canada, pp. 234-241, May 12-15, 1997.

[13: Men et al. 2011]

Hao Men, Biruk Gebre and Kishore Pochiraju, “Color Point Cloud Registration with 4D ICP Algorithm,” in Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Shanghai, pp. 1511-1516, May 9-13, 2011.

[14: Makadia et al. 2006]

Ameesh Makadia, Alexander Patterson and Kostas Daniilidis, “Fully Automatic Registration of 3D Point Clouds,” in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 1297-1304, June 17-22, 2006.

[15: Arun et al. 1987]

K. S. Arun, T. S. Hung and S. D. Blostein, “Least-squares Fitting of Two 3-D point Sets,” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 9, no. 5, pp. 698-700, September 1987.

[16: Scaramuzza et al. 2011]

Davide Scaramuzza and Friedrich Fraundorfer, “Visual Odometry [Tutorial],” IEEE Robotics and Automation Magazine, vol. 18, no. 4, pp. 80-92, December 2011.

[17: Nister et al. 2004]

David Nister, Oleg Naroditsky and James Bergen, “Visual Odometry,” in

Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp. 652-659, June 27-July 2, 2004.

[18: Kitt et al. 2010]

B.Kitt, A. Geiger and H. Lategahn, "Visual Odometry Based on Stereo Image Sequences with RANSAC-based Outliers Rejection Scheme," in Proceedings of IEEE Intelligent Vehicles Symposium, San Diego, USA, pp. 486-492, June 21-24, 2010.

[19: Jachalsky et al. 2010]

Jorn Jachalsky, Markus Schlosser and Dirk Gandolph, "Confidence Evaluation for Robust, Fast-Converging Disparity Map Refinement," in Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Suntec City, pp. 1399-1040, July 19-23, 2010.

[20: Lowe 2004]

David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, January 2004.

[21: Dalal et al. 2005]

Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, vol. 1, pp. 886-893, June 25,

2005.

[22: Saravanakumar et al. 2010]

S. Saravanakumar, A.Vadivel and C.G Saneem Ahmed, "Multiple Human Object Tracking using Background Subtraction and Shadow Removal Techniques," in Proceedings of IEEE International Conference on Signal and Image Processing (ICSIP), Chennai, pp. 79-84, December 15-17, 2010.

[23: Lee et al. 2003]

Dar-Shyang Lee, Jonathan J. Hull and Berna Erol, "A Bayesian Framework for Gaussian Mixture Background Modeling," in Proceedings of IEEE International Conference on Image Processing , vol. 3, pp. 973-976, September 14-17, 2003.

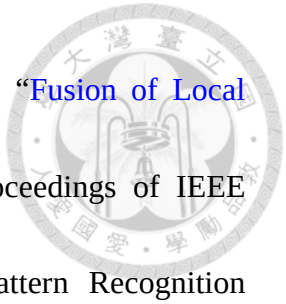
[24: Barnich et al. 2011]

Olivier Barnich and Marc Van Droogenbroeck, "ViBe: A Universal Background Subtraction Algorithm for Video Sequences," IEEE Transactions on Image Processing, vol. 20, no. 6, pp. 1709-1724, June 2011.

[25: Enzweiler et al. 2009]

Markus Enzweiler and Dariu M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no.12, pp. 2179-2195, December 2009.

[26: Tang et al. 2008]



Feng Tang, Michael Harville, Hai Tao and Ian N. Robinson, “Fusion of Local Appearance with Stereo Depth for Object Tracking,” in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, pp. 1-8, 23-28 June, 2008.

[27: Labayrade et al. 2002]

Raphael Labayrade, Didier Aubert and Jean-Philippe Tarel, “Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through “V-disparity” Representation,” in Proceedings of IEEE Intelligent Vehicle Symposium, vol.2, pp. 646-651, June 17-21, 2002.

[28: Hu et al. 2005]

Zhencheng Hu, Francisco Lamosa and Keiichi Uchimura, “A Complete U-V-Disparity Study for Stereovision Based 3-D Driving Environment Analysis,” in Proceedings of IEEE International Conference on 3-D Digital Imaging and Modeling, pp.204-211, June 13-16, 2005.

[29: Perrollaz et al. 2012]

Mathias Perrollaz, John-David Yoder, Amaury Negre, Anne Spalanzani and Christian Laugier, “A Visibility-Based Approaching for Occupancy Grid Computation in Disparity Space,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 3, pp. 1383-1393, September 2012.

[30: Perrollaz et al. 2010]

Mathias Perrollaz, Anne Spalanzani and Didier Aubert, "Probabilistic Representation of the Uncertainty of Stereo-Vision and Application to Obstacle Detection," in Proceedings of IEEE Intelligent Vehicles Symposium, San Diego, USA, pp. 313-318, June 21-24, 2010.

[31: Oniga et al. 2010]

Florin Oniga and Sergiu Nedevschi, "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection," IEEE Transactions on Vehicular Technology, vol. 59, no. 3, pp. 1172-1182, March 2010.

[32: Viola et al. 2003]

Paul Viola, Michael J. Jones and Daniel Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," in Proceedings of IEEE International Conference on Computer Vision (ICCV), Nice, France, vol. 2, pp. 734-741, October 13-16, 2003.

[33: Enzweiler et al. 2008]

M. Enzweiler, P. Kanter and M. Gavrila, "Monocular Pedestrian Recognition Using Motion Parallax," in Proceedings of IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, pp. 792-797, June 4-6, 2008.

[34: Wolf et al. 2004]

Denis Wolf and Gaurav S. Sukhatme, “Online Simultaneous Localization and Mapping in Dynamic Environments,” in Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, vol. 2, pp. 1301-1307, April 26-May 1, 2004.

[35: Danescu et al. 2012]

Radu Danescu, Cosmin Pantilie, Florin Oniga, and Sergiu Nedevschi, “Particle Grid Tracking System Stereovision Based Obstacle Perception in Driving Environments,” IEEE Transactions on Intelligent Transportation Systems Magazine, vol. 4, no. 1, pp. 6-20, January 26, 2012.

[36: Barth et al. 2009]

Alexander Barth and Uwe Franke, “Estimating the Driving State of Oncoming Vehicles From a Moving Platform Using Stereo Vision,” IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 4, pp. 560-571 , December 2009.

[37: Nedevschi et al. 2007]

Sergiu Nedevschi, Corneliu Tomiuc and Silviu Bota, “Stereo-Based Pedestrian Detection for Collision Avoidance Applications,” IEEE Transactions on Intelligent Transportation System, vol. 10, no. 3, pp. 380-391, September 2009.

[38: Krotosky et al. 2007]

Stephen J. Krotosky and M.M. Trivedi, “On Color-, Infrared-, and

Multimodal-Stereo Approaches to Pedestrian Detection", IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 4, pp.619-629, December 2007.

[39: Li et al. 2009]

Liyuan Li, Jerry Kah Eng Hoe, Shuicheng Yan and Xinguo Yu, "ML-Fusion based Multi-Model Human Detection and Tracking for Robust Human-Robot Interfaces," in Proceedings of IEEE International Workshop on Applications of Computer Vision (WACV), Snowbird, UT, pp. 1-8, December 7-8, 2009.

[40: Zitnick et al. 2002]

C. Lawrence Zitnick and Takeo Kanade, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 7, pp. 675-684, July 2000.

[41: Comaniciu et al. 2003]

Dorin Comaniciu, Visvanathan Ramesh and Peter Meer, "Kernel-Based Object Tracking," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, May 2003.

[42: Steder et al. 2011]

Bastian Steder, Radu Bogdan Rusu, Kurt Konolige and Wolfram Burgard, "Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries," in Proceedings of IEEE International Conference on Robotics and Automation

(ICRA), Shanghai, pp. 2601-2608, May 9-13, 2011.

Websites

[43: SIFT Keypoint Detector from David Lowe 2013]

SIFT Keypoint Detector. (2005, July). In David Lowe Personal Page. Retrieved April 3, 2013, from http://www.cs.ubc.ca/~lowe/keypoints/

[44: Kalman Filter from website 2013]

Kalman Filter Toolbox for MATLAB. (2004, June 7). In UBC. Retrieved June 6, 2013, from http://www.cs.ubc.ca/~murphyk/Software/Kalman/kalman.html

[45: HSL and HSV from wiki 2013]

HSL and HSV. (2013, June 6). In Wikipedia. Retrieved June 6, 2013, from http://en.wikipedia.org/wiki/HSL_and_HSV

[46: Connected-Component Labeling from wiki 2013]

Connected-Component Labeling. (2013, June 6). In Wikipedia. Retrieve June 6, 2013, from http://en.wikipedia.org/wiki/Connected-component_labeling

[47: Random Sample Consensus from wiki 2013]

RANSAC. (2013, May 13). In Wikipedia. Retrieved May 3, 2013, from http://en.wikipedia.org/wiki/RANSAC

[48: Interpolation from wiki 2013]

Interpolation. (2013, May 31). In Wikipedia. Retrieved May 31, 2013, from

http://en.wikipedia.org/wiki/Interpolation

[49: Accuracy For Stereo Vision from PointGrey 2010]

Article 63: How is depth determined from a disparity image? (2010, July 19). In PointGrey Official Knowledge Base. Retrieved May 31, 2013, from http://www.ptgrey.com/support/kb/index.asp?a=4&q=85

[50: UTE120 Combo ExpressCard from Uptech 2013]

UTE120 Combo ExpressCard. (2013, July). In Uptech Website. Retrieved July 30, 2013, from http://www.uptech.tw/product_detail.php?prod_id=488

[51: BumbleBee2 Product Datasheet from PointGrey 2013]

BumbleBee2 Documents- Product Datasheet. (2012, June). In PointGrey Official Website. Retrieved July 12, 2013, from http://www.ptgrey.com/products/bumblebee2/bumblebee2_xb3_datasheet.pdf

[52: URG-04LX-UG01 from Hokuyo]

Hokuyo URG-04LX-UG01 Documents- Product Datasheet. (2013, July 30). In Hokuyo Official Website. Retrieved July 30, 2013, from http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html

[53: SICK LMS100 from SICK]

SICK LMS100 Datasheet. (2013, July 30). In SICK Official Website. Retrieve July 30, 2013, from

http://www.sick-automation.ru/images/File/pdf/DIV05/LMS100_manual.pdf

[54: Point Cloud Library from PCL Website 2013]

Point Cloud Library. (2013, May 31). In PCL Website. Retrieved May 31, 2013, from http://pointclouds.org/

[55: NARF feature from PCL 2013]

NARF feature from Point Cloud Library. (2013, July 30). In PCL Website. Retrieved July 30, 2013, from

http://pointclouds.org/documentation/tutorials/narf_feature_extraction.php

[56: Rigid body from wiki 2013]

Rigid body. (2013, June 27). In Wikipedia. Retrieved June 27, 2013, from http://en.wikipedia.org/wiki/Rigid_body

[57: OpenCV from OpenCV official website 2013]

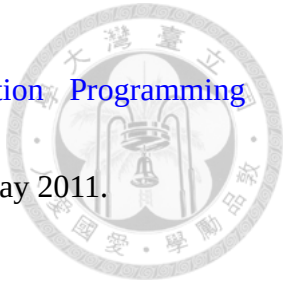
Open Source Computer Vision Library. (2013, June 27). In OpenCV official website. Retrieved June 27, 2013, from http://opencv.org/.

Books:

[58: Gonzalez & Woods 2008]

R. C. Gonzalez and R. E. Woods, Digital Image Processing, 3rd adapted ed., Editor: S. G. Miaou, Taiwan: Pearson, June 2008.

[59: Laganière 2011]



Robert Laganière, OpenCV 2 Computer Vision Application Programming Coolbook, 1st ed., Editor: Neha Shetty, Packt Publishing Ltd., May 2011.

[60: Spong 2005]

Mark W. Spong, Seth Hutchinson, M. Vidyasagar, Robot Modeling and Control, 1st ed., John Wiley & Sons, Inc., November 18, 2005

[61: Szeliski 2010]

Richard Szeliski, Computer Vision: Algorithms and Applications, 2011 ed., Springer, November 24, 2010.

[62: Thrun 2005]

S.Thrun, W. Burgard and D. Fix, Probabilistic Robotics, Editor: R. Arkin, London: The MIT Press, 2005.

[63: Buhmann 2003]

M. D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, 2003.

[64: Bradski et al. 2008]

Gary Bradski and Adrian Kaehler, Learning OpenCV, Editor: Mike Loukides, O'Reilly Media, Inc., 2008.