Vision based target tracking for autonomous vehicles

Jia, Zhen

2006

Jia, Z. (2006). Vision based target tracking for autonomous vehicles. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3466>

<https://doi.org/10.32657/10356/3466>

Nanyang Technological University

Downloaded on 26 Apr 2023 17:14:41 SGT

Vision based Target Tracking for Autonomous Vehicles

Jia Zhen

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2006

Statement of originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for higher degree to any other university or institution.

Date

Jia Zhen

To my parents.

Acknowledgement

This thesis would never have been completed if it were not for the amazing generosity, kindness and helpfulness of so many people.

I would like to begin by thanking my supervisor in Nanyang Technological University (NTU). Thank Dr. Arjuna P. Balasuriya for leading me and providing me a marvellous research environment, stimulating discussions, smiling with my ups and cheering me in my downs for the three years while I was in NTU. Thank Dr. Arjuna P. Balasuriya for his availability, encouragement and support during my postgraduate study in NTU. I also wish to thank him for his valuable comments and constructive criticisms on the draft of this thesis.

I would like to thank Prof. Wang Dan Wei, Head of Division of Control and Instrumentation at School of EEE, Nanyang Technological University. Thank him for his kind support and help during the three years while I was in NTU. He is also my supervisor during the last period of my Ph.D. candidature. Great thanks to him for his efforts coordinating my thesis's examination, correction and the final oral defense. Thanks to Ms. Lim Chew Mei and Ms. Lim Gim Hong for their help together with Prof. Wang on my PhD graduation and final oral defense.

I'd also like to thank Prof. Subhash Challa from University of Technology, Sydney (UTS), Australia for his valuable comments and suggestions during my Ph.D. study period. I thank him for his great support for my visiting scholar trip to UTS. It was really a wonderful time to work with him in UTS. I learn a lot from him. He is a nice and smart person with very good knowledge and research experience.

Thanks to Prof. W. Sardha Wijesoma, Prof. Soh Yeng Chai, Prof. Martin Adams,

Prof. Xie Li Hua, Prof. Chu Yun Chung, Prof. Chua Chin Seng for their support, encouragement and kind care during my NTU three years.

Thanks to Prof. Gamini Dissanayake from University of Technology, Sydney, Australia. Thank him for his kind help when I stayed as the visiting scholar in ARC Center of Excellence for Autonomous Systems in UTS. Also thanks to Dr. Kodagoda Ranadheera Sarath Kodagoda from UTS for his help when I was there.

Thanks to Prof. Er Meng Joo, Director of Intelligent Systems Center, Nanyang Technological University. Thank him for providing me the part-time job opportunity to work at Intelligent Systems Center and Singapore Technologies Engineering, Kinematics.

My thanks also go to Mr. Chia, Mr. Tee, Mr. Mok, Mr. Tan and Mr. Sow and all the other technical staffs of the Intelligent Robotics Lab, M&E Workshop, Intelligent Machine Research Lab and First Year Lab for their valuable and continuous support.

Thanks to Mr. Zhao Yu for helping me print the revised thesis and submit it.

I also wish to thank my friends and colleagues in NTU. They are Mr. Thomas Ang Yong Lim, Mr. Bharath Kalyan, Mr. Zheng Junjiang, Mr. Tang Fan, Mr. Zhang Bin, Mr. Wang Xiao, Mr. Wang Yigang, Mr. Tang Yu, Mr. Zhao Yu, Mr. Wang Dequn, Mr. Sun Yeow Cheng, Mr. Chen Cheng, Mr. Liu Chao, Mr. Liu Shuyong, Mr. Liu Nan, Mr. Zhou Weiqing, Mr. Wang Zhe, Miss Zhu Xiaoling, Miss Zhou Ji, Mr. Perera Linthothage Dushantha Lochana, Mr. Liu Xiaohui, Mr. Li Jiang, Mr. Li Xuchun, Mr. Zhang Wenbo, Dr. Wang Lei, Mr. Liu Bingbing, Mrs. Zhang Sen, Mr. Ebi Jose, Dr. Liu Zuojun, Miss Teoh Julnee, Miss Ong Yiqing, Miss Zhao Yufeng, Mr. Zhou Hao, Mr. Zhang Runxuan, Mr. Tang Ke, Mr. Chia Jin Hao, Mr. Qin Jing, Mr. John Mullane, Dr. Chen Binjing, Mr. Chen Ping, Mr. Wu Zhiguang and so many others. I would like to thank all the friends, students and staffs I know in Nanyang Technological University for assisting me on various occasions.

I must also thank Mr. Rajib Chakravorty from UTS. During my visiting scholar period at UTS, we worked together and had a lot of good memories. Rajib is a nice guy and a truly good friend.

My great thanks to Miss Fang Shenglan and Mr. Chen Shengsheng. Thank them

(especially Miss Fang Shenglan) for spending a lot of time correcting my thesis. Your help greatly improves the English quality of this thesis.

Thanks to the Nanyang Technological University and Singapore for providing me the financial support and a conducive atmosphere to carry out my Ph.D. research work.

Thanks to all the three reviewers for their valuable comments to improve the quality of this thesis.

My dearest father Jia Hongan and my dearest mother Liu Jingqi deserve the most special thanks for their understanding and encouragement. I would not be writing this if it were not for their support, understanding and love for the past 25 years. Thank you and I love you both.

Summary

The demand for fully Autonomous Guided Vehicles (AGV) for land, aerial and under-water environments is increasing day by day. However the applications of these vehicles are limited due to the lack of their on-board intelligence. This is mainly due to the limitations of the sensors used and the real-time processing capabilities for interpreting the captured sensor data and the knowledge of how to process the data correctly in the first place. Vision sensors are commonly used on AGVs. The amount of information available in visual data is very high compared to other sources of data. The interpretation of the data for useful navigation information is a challenging task. Therefore it is necessary to carry out research to recognize and interpret the visual data enabling the vehicle to obtain a better understanding of its working environment.

In many applications, it is important to track a visual target in consecutive image frames. Visual target tracking is one of the center research problems for vision based autonomous vehicle navigation. It is a challenging task when both the cameras and objects are moving in an outdoor unstructured environment. In this thesis, problems such as data fusion, image features extraction, region of interest identification, target state estimation and tracking have been closely studied. New methodologies have been developed for vision based target tracking for autonomous vehicles.

The major contribution of this thesis is the whole novel system proposed. In the proposed system, the sensor data fusion, image segmentation and clustering, template matching and updating and Extended Kalman Filter based target tracking are improved with the newly developed methods. All these contributions are explained briefly next.

First, in order that a moving object can be tracked in a moving background, the rela-

tive speed between the cameras and objects is estimated. This relative information is captured by optical flow vectors in the image sequence. In this thesis commonly used optical flow estimation methods are studied and Horn and Schunck's method is selected to estimate the optical flow vectors field. At the same time the visual depth information is estimated using the stereo disparity.

Next, in order to extract the region of interest (ROI) from the image, the image is clustered and segmented. Different visual features are fused to make the image segmentation and clustering more accurate to find the major motion regions. The K-means based and the novel unsupervised Bayesian on-line learning based image clustering algorithms are developed in this thesis. Video images segmentation results from different methods are also compared. Since the unsupervised Bayesian on-line learning based method gives better performance than the other methods tested in this thesis, it is used in the proposed visual target tracking system.

Once the image is segmented, the ROI needs to be localized continuously through the whole image sequence. First the Sum of Squared Differences (SSD) (or correlation) based template matching method is used to identify the region of interest from the clustered images. The matching is between the optical flow vectors template and the clustered optical flow vectors field. Next, in order to improve the matching performance, a novel Bayesian probability fusion based template matching and updating algorithm is developed for the proposed visual target tracking system. Different template matching methods are combined together through the Bayesian probability theory to give more accurate and robust matching performance.

After the visual features of interest are extracted from the image sequence, it is necessary to estimate the target's 3D position and velocity in the world coordinate frame for tracking. Here, the target's optical flow vectors, image color information, 2D spatial locations, and stereo disparities are used as visual features while the vehicle's inertial measurements are used to estimate the motion of the cameras. Then a novel sensor data fusion scheme is proposed. The visual information is fused with the vehicle's inertial motion sensors data based on the image processing and velocity compensation methods. The 3D dynamic information of the target is estimated in the world coordinate system

and later tracked by using an Extended Kalman Filter.

In this thesis the Extended Kalman Filtering is developed by integrating the Interacting Multiple Models (IMM) and Out-Of-Sequence Measurements (OOSM) solution algorithms to improve the tracking performance. First, since the normal linear dynamic models (such as the constant velocity model, constant acceleration model and so on) are not very accurate in target tracking, several simple and basic linear dynamic models are combined through the IMM method to approximate the unpredictable, complex or highly nonlinear dynamics of the moving target for Extended Kalman Filtering. Secondly, because the tracking system is designed based on multi-sensor data fusion, the OOSM problem exists. In this thesis the “1-step-lag” OOSM solution is conducted to solve the OOSM problem. With the IMM approach and OOSM solution, the target tracking accuracy is improved and the target’s 3D dynamic state estimation becomes more robust.

In this thesis the proposed visual target tracking scheme has been tested and its performance has been demonstrated through extensive experimental results. It can be concluded that the proposed system is suitable for autonomous vehicle visual target tracking.

KEY WORDS: Autonomous Guided Vehicle (AGV), Autonomous Underwater Vehicle (AUV), Bayesian Learning, Charged Coupled Device (CCD) Cameras, Data Fusion, Dynamics Model, Extended Kalman Filtering, Expectation-Maximization (EM), Gaussian Mixture Models (GMMs), Gradient Descent, Image Segmentation and Clustering, Interacting Multiple Models (IMM), K-Means, Kalman Filtering, Kinematic Model, Maximum A Posteriori Probability (MAP), Maximum Likelihood Estimation, Moving Camera and Moving Object (MCMO), Occlusion, Optical Flow, Out-Of-Sequence Measurements (OOSM), Pinhole Camera Projection, Probability Distribution Function (PDF), Sensor Fusion, Sum of Squared Differences (SSD), Sum of Squared Errors (SSE), Sum of the Absolutely RGB Difference (SAD), Stereo Vision, Target Tracking, Template Matching and Updating, Unsupervised Learning

Table of Contents

Acknowledgements	i
Summary	iv
List of Figures	xiii
List of Tables	xxiv
1 Introduction	1
1.1 Motivation	1
1.2 Applications of Visual Target Tracking for Autonomous Vehicle Navigation	3
1.3 Objective	6
1.4 Main Contributions	6
1.5 Thesis Outline	9
2 Literature Review	12
2.1 Introduction	12
2.2 Vision based Target Tracking for Autonomous Vehicle Navigation	13
2.2.1 Visual Target Tracking for Autonomous Land Vehicle Navigation	13

TABLE OF CONTENTS

viii

2.2.1.1	Visual Landmark Tracking for Autonomous Land Vehicles Navigation	13
2.2.1.2	Human Following for Autonomous Land Vehicle Navigation	14
2.2.1.3	Visual Target Tracking for Autonomous Land Vehicle Localization and Map Building	17
2.2.1.4	Visual Target Tracking with Pan-Tilt Camera Platforms in Autonomous Land Vehicle	18
2.2.1.5	Visual Target Tracking for Multiple Mobile Robots Cooperation	19
2.2.1.6	Other Visual Target Tracking Methods for Autonomous Land Vehicle Navigation	21
2.2.2	Visual Target Tracking for Autonomous Underwater Vehicle	23
2.2.3	Visual Target Tracking for Autonomous Aerial Vehicle	24
2.3	Visual Target Tracking Technologies	26
2.3.1	Motion Detection	26
2.3.2	Target Tracking	28
2.3.3	Optical Flow Vectors Based Target Tracking Systems	33
2.4	Multiple Sensor Data Fusion	35
2.4.1	Data Fusion Methodologies	36
2.4.2	Applications of Data Fusion for Autonomous Vehicles	38
2.4.2.1	Fusion based Visual Target Tracking for Autonomous Vehicle Navigation	39
2.5	Summary	41
2.5.1	Remaining Research Challenges	44

TABLE OF CONTENTS

ix

3	Proposed Tracking System	47
3.1	Introduction	47
3.2	Proposed Tracking System	48
4	Visual Feature Extraction	52
4.1	Optical Flow Estimation	52
4.1.1	Horn and Schunck's Method	53
4.1.2	Lucas and Kanade's Method	55
4.1.3	The Three-Dimensional (3D) Structure Tensor Technique for Optical Flow Estimation	56
4.2	Visual Depth from Stereo Imaging	61
4.2.1	Introduction	61
4.2.2	Depth from Stereo	63
4.2.3	Disparity Estimation	64
4.2.4	Some Stereo Images	65
4.3	Experimental Results of Optical Flow Estimation	66
4.4	Summary	70
5	Visual Features Classification: Image Segmentation and Clustering	74
5.1	Introduction	74
5.2	Image Clustering and Segmentation Using the K-means Algorithm	77
5.2.1	K-means Algorithm	78
5.2.2	K-means Cluster Validity Measures	79
5.2.2.1	Description of the K-means Cluster Validity Measures Method	81

TABLE OF CONTENTS

x

5.3	Fusion Based Video Images Segmentation Using Unsupervised Bayesian On-line Learning	83
5.3.1	Introduction	83
5.3.2	Probability Based Multiple Visual Features Fusion Segmentation Theory	87
5.3.2.1	Different Image Visual Features	87
5.3.2.2	Multiple Visual Features Fusion for Probability based Video Images Segmentation	89
5.3.2.3	Weighting Algorithm for Probability based Multiple Visual Features Fusion	90
5.3.3	Initial Image Segmentation Process	92
5.3.4	Unsupervised Bayesian On-line Learning for Gaussian Distribution Case	93
5.3.4.1	The Unsupervised Bayesian On-line Learning Algorithm	95
5.4	Experimental Results	98
5.4.1	K-means Image Segmentation Results	98
5.4.2	Unsupervised Bayesian On-line Learning based Video Images Segmentation Results	100
5.5	Summary	114
6	Visual Features Localization: Template Updating and Matching	118
6.1	Normal Template Matching Method	119
6.1.1	SSD (Correlation) based Template Matching	119
6.1.1.1	Advantages and Shortcomings of the Correlation based Template Matching	120

TABLE OF CONTENTS

xi

6.2	Bayesian Probability Fusion based Template Matching and Updating . .	122
6.2.1	Introduction	123
6.2.2	Bayesian Probability based Two Template Updating and Match- ing Algorithms Fusion	127
6.2.2.1	Bayesian Probability based Template Matching Algorithms Fusion	127
6.2.2.2	Sum of Squared Errors Template Matching	129
6.2.2.3	Gaussian Mixture Models for Template Matching	133
6.2.2.4	PDF Parameters Initialization for GMMs	134
6.2.3	Minimization of the Bayesian Fusion Equation	136
6.2.4	Template Updating Strategy	137
6.2.5	Initial Template Matching	138
6.3	Experimental Results	139
6.4	Summary	148
7	Target Tracking System	150
7.1	Sensor Data Fusion based Target 3D Dynamics Information Estimation .	150
7.2	Extended Kalman Filter based Target Tracking System	156
7.3	IMM Approach for Target Tracking	158
7.3.1	Introduction	158
7.3.2	Linear Dynamic Models for Extended Kalman Filter	159
7.3.3	Interacting Multiple Models Approach	161
7.3.4	Interacting Multiple Models For Extended Kalman Filter Estimation	163
7.4	The OOSM Problem and Solution for Extended Kalman Filtering	164

TABLE OF CONTENTS

xii

7.4.1	Introduction	164
7.4.2	Formulation of the OOSM Problem	167
7.4.3	Summary of The OOSM Solution Algorithm	169
7.5	Summary	171
8	Target Tracking Experimental Results	172
8.1	Noise Identification for the Tracking System	173
8.1.1	Noise Identification for the Discrete State Space Model	174
8.1.2	Measurement Noise	176
8.1.3	Noise Initialization of the State Estimator	176
8.1.4	Root-Mean-Square Error	177
8.2	Experimental Results in the Structured Environment	178
8.3	Experimental Results in the Outdoor Unstructured Environment	194
8.4	Experimental Results Discussion	200
9	Conclusions and Future Work	211
9.1	Conclusions	211
9.2	Future Work	217
	Author's Publications	221

List of Figures

1.1	Some applications of the visual target tracking system for autonomous vehicle navigation. Fig.1.1.1 shows the general high way system (the image taken from http : //www.tfhrc.gov/humanfac/presentation/). The visual target tracking based intelligent transportation system can be developed here for road safety, vehicle avoidances and so on. Fig.1.1.2 shows the mobile robots soccer (the image taken from the web site of Automation Laboratory at University of Mannheim). Fig.1.1.3 shows the human assistance mobile robots (the image taken from the proceedings cover of IEEE ICRA 2005). The visual target tracking can help the mobile robots to fulfill the tasks in all these applications.	3
1.2	Some applications of the visual target tracking system for AUV navigation. Fig.1.2.1 shows the test bed AUV (Twin-Burger 2) for the underwater cabal following (the image taken from [4]). Fig.1.2.2 shows the underwater images of AUV docking (the image taken from [222]).	5
3.1	The Proposed Tracking System	50
4.1	The Simple Camera Geometry for Stereo Images from Horn's book [71] .	63
4.2	Some Stereo Images in Both the Indoor and Outdoor Structured and Unstructured Environments. These images are used later to test the proposed target tracking system in Chapter 8.	65

4.3	The distance estimation result. This figure shows the estimated visual depth vs. the frame number for the stereo images of Fig.(4.2.1) and Fig.(4.2.2). The x axis is the image frame number. The y axis is the estimated distance (cm) between the cameras and the object.	66
4.4	One person moving in the outdoor unstructured environment. The first three figures are the color images. Fig.(4.4.4) is the gray level image for the optical flow estimation.	67
4.5	Optical flow vectors fields estimated from Lucas and Kanade's method .	68
4.6	Optical flow vectors fields estimated from Horn and Schunck's method. .	68
4.7	Indoor moving target with moving background. A blue color object is moving in the image sequence. It can be seen that the background is changing frame by frame.	69
4.8	Optical flow vectors fields from the image sequence of Fig.(4.7) with the modified Horn and Schunck's method. Figures (4.8.3) and (4.8.4) are two of the enlarged ones for clarity.	70
4.9	The black area is moving in the images to test the optical flow estimation algorithms.	71
4.10	Optical flow vectors estimated from the structure tensor method for the image sequence of Fig.(4.9).	71
4.11	Optical flow vectors estimated from Horn and Schunck's Method for the Image Sequence of Fig.(4.9).	72
5.1	Moving object with moving background in the indoor situation	99
5.2	The motion vectors field segmentation and clustering results from the image thresholding and the K-means clustering.	99
5.3	The image clusters' PDF parameters initialization (mean) with the EM algorithm for the Fig.(5.1) segmentation.	102

5.4	The image clusters' PDF parameters initialization (variance) with the EM algorithm for the Fig.(5.1) segmentation.	103
5.5	The image clustering and segmentation results without the Bayesian learning. The results shown here correspond to the original figures in Fig.5.1.	104
5.6	The image clustering and segmentation results with the unsupervised Bayesian on-line learning (5 Clusters). The results shown here correspond to the original figures in Fig.5.1.	104
5.7	The image segmentation results with the unsupervised Bayesian on-line learning. Only one cluster from the clustered images is shown in these figures. The results shown here correspond to the original figures in Fig.5.1.	105
5.8	Image clustering and segmentation results with 3 clusters. The results shown here correspond to the original figures in Fig.5.1.	105
5.9	The image visual features clusters' parameters learning (Mean) for the Fig.(5.1) segmentation.	107
5.10	The image visual features clusters' parameters learning (Variance) for the Fig.(5.1) segmentation.	108
5.11	The image sequence with a car driving along the road. The background is changing continuously.	109
5.12	The car driving image sequence with 3 clusters segmentation results. The results shown here correspond to the original figures in Fig.5.11.	109
5.13	Two persons are moving towards each other.	110
5.14	The two persons moving image sequence segmentation results without the unsupervised Bayesian on-line learning. The results shown here correspond to the original figures in Fig.5.13.	110
5.15	The two persons moving image sequence segmentation results with the unsupervised Bayesian on-line learning. The results shown here correspond to the original figures in Fig.5.13.	111

5.16	A blue object moving out of the image scene.	112
5.17	The image segmentation results with the unsupervised Bayesian on-line learning when the object leaves the image view. The results shown here correspond to the original figures in Fig.5.16.	112
5.18	The object is occluded by another object.	113
5.19	The image segmentation results with the unsupervised Bayesian on-line learning when the object is occluded by the other object. The results shown here correspond to the original figures in Fig.5.18.	113
6.1	Correlating an Image With Noise from [226]	121
6.2	The template from the clustered optical flow vectors field and the Euclidean distance based matching result. The smallest value in Fig.6.2.2 indicates the template's best matched region in the images.	140
6.3	The template matching results when the moving object is occluded by another stationary object. The blue object in the first frame is considered as the initial template.	141
6.4	The template matching results when the object is occluded by another moving object. The blue object in the first frame is considered as the initial template.	141
6.5	The template matching results when the black object is moving towards the camera. Here the template is not updated with the Bayesian fusion algorithm developed in this Chapter. Thus finally the template drifts away. The matching results are not good. The black object in the first frame is considered as the initial template.	142
6.6	The template matching results when the object is moving towards the camera. The template matching performance is improved with the proposed robust template updating algorithm.	142

- 6.7 The template matching results when the object's shape has some changes. Here the black target is rotating greatly in two different directions. The black object in the first frame is considered as the initial template. The proposed template matching algorithm is to locate the template's most matched region in the images when the object has great rotations. The proposed Bayesian fusion based method gives good result. 143
- 6.8 The template matching results when the object has some rotations. Compared with Fig.6.7, the template matching performance is bad if the template is not matched and updated by the algorithm developed in this Chapter. 144
- 6.9 The template matching results when the blue object leaves the image scene. The blue object with its full shape from the first frame is considered as the initial template. The template matching performance is good with the proposed method. The template matched region is changing due to the changes of the object's size in the image scene. 145
- 6.10 The person is moving with a moving camera. The person in the first image frame is considered as the initial template. The proposed Bayesian fusion based algorithm is used to locate the person in the image sequence. Through the whole image sequence the appearance and size of the target person in the images change a lot. Here with the proposed method the template is updated robustly and the person is correctly tracked. The template's size is scaled due to the distance changes between the person and the camera. 145
- 6.11 The template matching result when two persons are moving in the cluttered outdoor environment. The two persons are moving around each other and one of them is tracked. Here the tracker is lost when the person is occluded by another one (Fig.(6.11.2)). With the proposed Bayesian fusion based algorithm, when the occlusion is finished, the template is updated together with the first template and after the occlusion, the previously tracked person is still correctly tracked. 146

7.1	Experiment Setup with the Target's 3D World Coordinate Velocity Estimation Procedures	151
7.2	The Pinhole Camera Projection Model	152
7.3	Kinemics Model of Autonomous Vehicle	153
8.1	The moving platform with different sensors. In this experiment the moving platform is designed based on a man-push trolley. On this trolley, two PCs (industry PC and embedded PC) and three sensors (AHRS, tachometers and CCD Cameras) are mounted together to do sensors data capturing and processing for the target tracking system.	173
8.2	The image sequence of a moving target with the moving background in a structured indoor environment. A blue color object is moving from left to right in the image sequence. It can be seen that the background is changing frame by frame.	178
8.3	The stereo images and distance estimation results. One pair of stereo images are shown in Fig.(8.3). Fig.(8.3.2) shows the estimated visual depth vs. the frame number. In Fig.(8.3.2) the x axis is the image frame number. The y axis is the estimated distance (cm) between the cameras and the object.	179
8.4	The motion sensors data and cameras' platform movement trajectory in the 2D plain. Fig.(8.4.1) shows the estimated two wheels velocities. Here the x axis is the frame number. The y axis is the velocity (cm/s). From the estimated velocity of the moving platform, the platform's movement trajectory can be plotted as shown in Fig.(8.4.2). The x and y axes are the 2D positions in the plain (cm).	180
8.5	The target's 3D tracking results without template updating in the structured indoor environment. The target tracking results are not good in these figures.	182

- 8.6 The target tracking results with template updating in the structured indoor environment. The target's $3D$ position is projected into the $2D$ images to show the performance. The IMM method is implemented with the Extended Kalman Filter to improve the tracking performance. The size of the template is changing due to the distance changes. The rectangle can accurately fit the target region which shows the tracking system performs well. 182
- 8.7 The target position tracking results in the $2D$ images. The target's $3D$ position is projected into the $2D$ images to show the performance. In Fig.8.7 the measurement target $2D$ position and the estimated target $2D$ position along the trail are compared. 183
- 8.8 Target's position and velocity estimation results from different models. The estimates along the x dimension of the target's $3D$ world coordinate position and velocity are shown here. The estimation results with the IMM algorithm are relatively more accurate than those from other models. In this figures the target position measurements are the ground truth values. 185
- 8.9 Multiple repetitions of the experiments are performed and the RMS errors are compared. The estimates along the x dimension of the target's $3D$ world coordinate position and velocity are shown here. Here the RMS errors from the estimation results with the IMM algorithm are relatively smaller. 186
- 8.10 Target dynamic information estimation results from different models when the target is moving with a direction turn. The estimates along the x dimension of the target's $3D$ world coordinate position and velocity are shown here. In such a situation, the estimation results with the IMM algorithm are still quite accurate. In these figures the target position measurements are the ground truth values. 187

8.11	Multiple repetitions of the experiments are performed and the RMS errors are compared. The estimates along the x dimension of the target's 3D world coordinate position and velocity are shown here. Here the RMS errors from the estimation results with the IMM algorithm are relatively smaller.	188
8.12	Target dynamic information estimation results with and without the OOSM solution. One dimensional data along the x axis of the target's 3D world coordinate position and velocity are shown here. The estimation results with the OOSM solution are relatively more accurate than those without the OOSM solution. In this figures the target position measurements are the ground truth values.	190
8.13	Multiple repetitions of the experiments are performed and the RMS errors are compared. One dimensional data along the x axis of the target's 3D world coordinate position and velocity are shown here. Here the RMS errors from the estimation results with the OOSM solution are relatively smaller.	191
8.14	Target dynamic information estimation results with and without the OOSM solution when the target is moving with a direction turn. One dimensional data along the x axis of the target's 3D position and velocity are shown here. In such a situation, the estimation results with the OOSM solution are still quite accurate. In these figures the target position measurements are the ground truth values.	192
8.15	Multiple repetitions of the experiments are performed and the RMS errors are compared. One dimensional data along the x axis of the target's 3D position and velocity are shown here. Here the RMS errors from the estimation results with the OOSM solution are relatively smaller.	193
8.16	Two persons occluded each other and the segmented optical flow vectors fields for this image sequence.	195
8.17	The tracked target is missed when the two persons cross each other.	196

8.18	Tracking with a dynamic model direction prediction to solve the occlusion problem.	196
8.19	Tracking performance with the modification for the occlusion situation. .	197
8.20	Two persons are moving with a moving camera in a cluttered environment. Here when these two persons are moving across each other, the tracker is still good. The person from left to right is tracked through the whole image sequence. There is no missed tracking problem at the occlusion region. The tracking system can accurately track the person when there are several similar motion regions in the image. The occlusion problem is solved using the target's estimated motion direction in the image sequence.	197
8.21	The target 3D tracking results when a person is moving in the cluttered environment. Here the camera is rotating from left to right. This is considered as the cameras' both rotational and translational motions situation. The person is correctly tracked with the proposed system.	198
8.22	The target 3D tracking results when a person is moving in the outdoor unstructured and cluttered environment. The person is self-rotationally moving and the cameras are also moving. The person's visual features are changing through the image sequence. Here the person is correctly tracked with its visual features' changes.	199
8.23	The target 3D tracking results when the person is moving in the cluttered outdoor environment. The person is moving from the dark side to the bright side. In such a situation the visual features of the person are changing. The person looks much brighter frame by frame. If the tracking is done without the motion based algorithm developed in this thesis, the tracking results are not good and the tracker is lost (Fig.(8.23.4) and Fig.(8.23.5)). Here the performance is improved with the proposed motion based tracking algorithm as shown in Fig.(8.23.6), Fig.(8.23.7), Fig.(8.23.8), Fig.(8.23.9) and Fig.(8.23.10). The person is accurately tracking with its various visual features.	200

- 8.24 Two persons are moving with the moving cameras in a cluttered environment. Here when the first person is fully occluded by the second person, the tracker is lost. With the proposed Bayesian fusion based template matching and updating algorithm in Chapter 6, when the occlusion is finished, the tracking template is updated together with the first template and the right person is still being tracked. In such a situation the tracking algorithm still can give good performance. 201
- 8.25 The target's 3D tracking results when three persons are moving in the cluttered outdoor environment. One person is correctly tracked. However when the tracked person leaves the image scene, the tracker is lost. Since the proposed method is based on the motion features tracking, then the algorithm tracks the person with the most similar motion instead. (Fig.(8.25.6) and Fig.(8.25.7)) 202
- 8.26 The target tracking results when several different people enter the image scene. The cameras are also moving in an indoor environment. Here the tracker is initialized when one person is entering the image scene and the tracker is terminated when this person is leaving the image scene. 202
- 8.27 The target tracking results when several people are moving in the same environment. The cameras are also moving. As same as Fig.8.26, the tracker can be automatically initialed and terminated for each person. Here the tracking can be accurately done when several people move in the same environment. Even if two persons have quite similar visual properties (Fig.(8.27.8), Fig.(8.27.9) and Fig.(8.27.10)), the tracking can still be accurately done. 203
- 8.28 The person's estimated 3D position in the world coordinate. Fig.(8.28.1) shows the person's projected 2D position in the $X - Y$ plain. Fig.(8.28.2) shows the person's projected 2D position in the $X - Z$ plain. 204

8.29	The person's estimated 3D position in the world coordinate. Here the person is moving with a direction tune not along a straight line as shown in the Fig.(8.28). Fig.(8.29.1) shows the person's projected 2D position in the $X - Y$ plain. Fig.(8.29.2) shows the person's projected 2D position in the $X - Z$ plain.	205
9.1	Generic Outdoor Mobile Explorer (GenOME) in NTU	219
9.2	NTU-UAV	219

List of Tables

2.1	The Classification of Data Fusion Algorithms from [129]	36
6.1	The comparison results of different template matching and updating methods. The proposed algorithm can achieve an average accuracy above 90% for the target tracking in different situations.	147

Chapter 1

Introduction

1.1 Motivation

At present, the usage of autonomous vehicles is growing especially in applications such as manufacturing, hazardous materials handling, surveillance, etc. The basic task in any such application is the perception of the environment through one or more sensors. Processing of the sensor input results in a particular representation of the unknown environment, which can then be used for navigating and controlling the vehicle. The autonomous vehicle navigation in an unknown environment is thus a quest that many researchers have tackled over the years.

The general sensors used for autonomous vehicles include infra-red, sonar, laser, radar and so on [134]. Compared to these types of sensors, vision sensors provide a whole new way for autonomous vehicles to image the environment [43] [54] [119] [174]. Video images plus the specialized computer vision algorithms can provide high resolution information concerning the shape or range of the near objects and the environment. Coupled with the availability of the increased computational power, visual sensor information not only becomes appealing but also easily attainable in real-time. Thus in this thesis the computer vision based navigation scheme is proposed for the Autonomous Guided Vehicle (AGV) navigation.

During the past ten years much research has gone into the area of computer vision

for autonomous vehicle navigation [43]. Many algorithms and methods have been proposed, all with an ultimate common goal: to give intelligence to autonomous vehicles to interpret the visual information. If the research goal is to send an autonomous vehicle from one coordinate location to another, there is sufficient accumulated expertise in the research community today to design algorithms which could do that in a typical environment. But if the goal is to carry out the function-driven navigation, such as chasing or following moving targets, avoiding the obstacle which is somewhere in a given hallway and stopping at a stop sign (e.g. docking) under varying illumination and background conditions, it is still eons away. It is still the central research problem for the vision based autonomous vehicle navigation that an autonomous vehicle must be aware of the position and dynamic information of the certain moving objects encountered in the environment.

Normally CCD cameras are used as vision sensors for autonomous vehicle navigation. CCD cameras' installation and maintenance costs are quite minimal and stereo CCD cameras can also provide the three-dimensional (3D) scene analysis [54]. The position and velocity of the target relative to the vehicle can be established continually by processing the stream of the camera images, and this information can be used to navigate the vehicle.

Generally the camera/object states in a tracking system can be divided into 4 categories [149]: 1)Stationary Camera, Stationary Object (SCSO), 2)Stationary Camera, Moving Object (SCMO), 3)Moving Camera, Stationary Object (MCSO), 4)Moving Camera, Moving Object (MCMO). In the case of visual target tracking by autonomous vehicles, both the camera and object move with respect to each other and it is the MCMO state.

Therefore, the main motivation of this research is to design a CCD camera based visual target tracking system for autonomous vehicle navigation in the MCMO state.

1.2 Applications of Visual Target Tracking for Autonomous Vehicle Navigation

The moving target's position and velocity information can aid the autonomous vehicle to determine what constitutes its surroundings and what actions if necessary are to be taken. The potential applications of such visual target tracking systems are: the autonomous vehicle navigation, map building, robot localization, path planning, obstacle avoidance, surveillance systems, intelligent transportation systems and human assistance mobile robots and so on [43].

1.1.1:

1.1.2:

1.1.3:

Figure 1.1: Some applications of the visual target tracking system for autonomous vehicle navigation. Fig.1.1.1 shows the general high way system (the image taken from [http : //www.tfhr.gov/humanfac/presentation/](http://www.tfhr.gov/humanfac/presentation/)). The visual target tracking based intelligent transportation system can be developed here for road safety, vehicle avoidances and so on. Fig.1.1.2 shows the mobile robots soccer (the image taken from the web site of Automation Laboratory at University of Mannheim). Fig.1.1.3 shows the human assistance mobile robots (the image taken from the proceedings cover of IEEE ICRA 2005). The visual target tracking can help the mobile robots to fulfill the tasks in all these applications.

In the intelligent transportation systems (Fig.1.1.1), the extraction and tracking of objects of interest are the necessary prerequisites for the development of intelligent autonomous vehicles or mobile robots. In the case of forward collision avoidance, visual moving objects tracking helps to distinguish the potential collision threats in terms of their relevance to the intended path of the vehicle [103]. Apart from such applications visual target tracking can also provide assistance to human drivers. One such application can be in drowsy driver warning systems. The knowledge of moving objects around

the vehicle enables a driver assistant system to alert a driver of the potential collisions and dangers [103]. Other applications in the intelligent transportation systems can be the design of an optimal trajectory for one vehicle under normal conditions, in order to overtake a single, slower-moving vehicle on a predetermined road [184].

Path planing for autonomous vehicles is to plan in real time a collision-free path in the presence of dynamically moving objects and with a limited sensing range [167]. Path planning is a fundamentally important issue in robotics, which also requires the autonomous vehicle to be aware of the objects in the environment. Complete coverage path planning (CCPP, which is also called region filling or area covering) of cleaning robots is a special type application of path planning in a two-dimensional (2D) environment. Here visual target tracking can help the robot pass through every area in the workspace and avoid the obstacles [227].

There are other applications where visual target tracking systems are used such as in the human assistance mobile robotics (Fig.1.1.3). Mobile robots equipped with mechanisms for communication, interaction, and behaviors are employed more and more outside traditional manufacturing applications, such as in museums or exposition areas [62] [83]. In order to operate autonomous robots in the same environment as a human, obstacle avoidance techniques in a dynamic environment are required. Such collision avoidance problems between human and robot can be called as interaction based on the position information [100], which can be solved by visual target tracking.

In security or surveillance applications, it is often more important to carefully position multiple sensors in order to cover a large structured environment. In this content, mobile robot-based trackers are attractive; they can potentially reduce the number of sensors needed in the tracking network, and they should be able to adapt to the movements of the targets or the dynamic changes in an environment by re-positioning themselves in responses [94].

In the underwater environment visual target tracking can be used for Autonomous Underwater Vehicle (AUV) navigation. For example in order to study the behavioral patterns of unknown underwater life forms, it is necessary to observe them carefully for a longer period. It is therefore important for AUVs to stay close to the object being

1.2.1:

1.2.2:

Figure 1.2: Some applications of the visual target tracking system for AUV navigation. Fig.1.2.1 shows the test bed AUV (Twin-Burger 2) for the underwater cabal following (the image taken from [4]). Fig.1.2.2 shows the underwater images of AUV docking (the image taken from [222]).

observed by moving with it such as the fish following. Also, for a reliable functioning of underwater man-made systems, it is necessary to maintain them by routine observations. Therefore, the underwater exploration and maintenance requires the AUV to have proper observations and as a result close tracking and following along or with these objects is necessary [222]. Arjuna and Tamaki [4] propose a sensor fusion technique for Autonomous Underwater Vehicle (the test bed vehicle Twin-Burger 2 in University of Tokyo (Fig.1.2.1)) to track and follow underwater cables through video images. Also Arjuna et. al. [5] propose the vision based tracking system for underwater docking. There are also many other robotics applications requiring the visual obstacle avoidance and/or object identification in unstructured underwater environments. For example the U.S. Navy is investigating the use of autonomous underwater vehicles to accurately detect and classify underwater mines prior to beach landings in the hostile territory [181].

Some other applications of visual target tracking systems for autonomous vehicle navigation can also be in the following areas:

- The mobile robots soccer [132] [191] (Fig.1.1.2).
- Obstacle avoidance for the robot manipulators in the factory automation doing repetitive and dull work [231].
- Obstacle avoidance for the robotic motion planning [156].

- The motion control of autonomous vehicles in the car manufacturing industry. Many car manufacturers plan to equip their vehicles in the near future with computer-aided visual target tracking capabilities for parallel parking or automatic stop-and-go mode in traffic jams [112].

1.3 Objective

It is clear that visual target tracking for autonomous vehicle navigation is a very important research issue. In order to successfully achieve all the applications, it is very useful and necessary for the autonomous vehicle to have the knowledge of the 3D dynamic information of the objects of interest in the environment.

Therefore the objective of this thesis is to develop a robust and efficient visual target tracking scheme for the AGV navigation by using their onboard sensors. The tracking system should be able to:

- handle the unknown or changing environmental conditions.
- track the object when both the cameras and objects are moving.
- track the objects in different states (different shape, color, texture, dynamics or any other properties) and without the specific prior visual information of the object.

Establishing a system that produces quality results while overcoming all the challenges implicated by operation in the real-time environment is the ultimate goal of the computer vision based autonomous vehicle target tracking research in this thesis.

1.4 Main Contributions

This thesis develops a novel and robust visual target tracking scheme for autonomous vehicle navigation. The proposed visual target tracking scheme solves the problems of tracking the moving object from the moving platform without knowing the prior visual features of the environment and target. The proposed tracking scheme successfully

demonstrates a whole new way to integrate different image processing, data fusion and state estimation algorithms to robustly extract the useful visual features, estimate the target's 3D velocity and position and finally track the target's dynamics in world coordinates. In detail, the contributions of this thesis can be divided into following aspects:

- In order to derive and track the moving target's 3D dynamics in world coordinates, a novel tracking scheme is developed mainly based on the image motion vectors. In such a way the target identification is independent of its specific visual properties, which makes the proposed system track objects under different states and environmental conditions.
- In order to accurately estimate the motion vectors from the images, three commonly used optical flow vectors estimation algorithms are studied and tested. Among those Horn and Schunck's algorithm is used for the proposed visual tracking scheme. These comparison studies and experimental results can be helpful for the further research work on optical flow.
- Since the object and background are both moving, there are many visual features in the image. Therefore it is necessary to classify these visual features and find the useful ones. In this thesis, data fusion based methods are proposed for image segmentation and clustering. Optical flow vectors are fused with the target's color and spatial location information, which greatly improve the segmenting and clustering performance. The segmentation of the image by visual data fusion also enables the general target identification when the optical features of the image are changing continuously or even when the object is moving in a very complex and cluttered environment or is occluded by other objects.
- In this moving object and moving cameras' tracking system, the visual features of the target and background are changing continuously. An unsupervised on-line learning algorithm based on the Bayesian estimation theory is proposed in a novel probabilistic way for image pixels clustering. With this on-line learning algorithm, the image classifier is robustly updated frame by frame and the image segmentation performance is greatly improved.

- In order to localize the features of interest in the classified images, a novel Bayesian probability based fusion algorithm is proposed for adaptive and accurate template matching. Here, two different template matching algorithms are weighted and combined by their matching probability functions to give a final robust matching technique. With the Bayesian fusion of two different template matching methods, a novel and robust template updating method is also proposed. Here the template is updated together with its Gaussian Mixture Models (GMMs) parameters and the matching functions' PDF parameters.
- After all the visual features and data are obtained from different sensors, the target's 3D dynamics is estimated. In this thesis, a novel data fusion algorithm is proposed for the target's 3D position and velocity estimation in the world coordinate system. The fusion algorithm is developed by fusing the target's visual features with the camera motion parameters based on the velocity compensation method. Later, with the Extended Kalman Filter measurement equation, data fusion based target dynamic tracking is achieved.
- Generally, the dynamic properties of natural moving objects are not simple or linear. In this thesis, a suitable and more flexible object dynamic model is proposed for the tracking system. Here several simple and basic linear dynamic models are combined through the IMM approach to approximate the target's complex, highly nonlinear or unknown dynamic properties. With the IMM approach the target dynamics tracking performance is improved.
- In this thesis the proposed system is developed based on sensor data fusion to estimate and track the target in the 3D world coordinate system. There are delayed measurement problems when all the data are fused in the fusion center. Here the 1-step-lag Out-Of-Sequence Measurements (OOSM) solution is introduced and integrated with Extended Kalman Filter in the proposed visual target tracking scheme. With the 1-step-lag OOSM solution, the tracking performance is improved and made robust.

1.5 Thesis Outline

- Chapter 1 provides the motivation and objective of this thesis. In this Chapter the major contributions are also concluded.
- Chapter 2 concisely studies and reviews the existing methods for the research topic of this thesis. Three related research fields (vision target tracking for autonomous vehicle navigation, visual target tracking technologies and sensor fusion methodologies) are reviewed. Justifications and conclusions are summarized and the remaining research challenges are presented.
- In Chapter 3, the overview of the proposed vision based target tracking scheme is presented. Based on literature reviews, this thesis proposes a new methodology to track the moving targets from a moving platform in an unknown environment. This Chapter explains the proposed scheme in details. The next chapters are organized following the flow chart of the proposed target tracking scheme.
- Chapter 4 describes the visual features extraction algorithms. These information are used in the following steps to estimate the 3D dynamics of the target for tracking. In this Chapter, first, three optical flow estimation algorithms are studied and tested. Experimental results compare the performance of different optical flow estimation methods. Next, the stereo disparity estimation method is described and used to find the target's 3D visual depth.
- Chapter 5 presents the image segmentation and clustering algorithms. In order to identify the major motion regions, the visual features are segmented. In this Chapter, fundamentals on visual features fusion based image segmentation and clustering techniques are presented. Then the simple K-means algorithm is developed for the fusion based image clustering. In addition, the K-means cluster validity measures method is introduced and it overcomes the limitation of having to indicate the number of clusters by incorporating a validity measure based on the intra-cluster and inter-cluster distance measures. Next, in details the unsupervised Bayesian on-line learning is developed and presented for the fusion based image segmentation. The unsupervised on-line learning based method updates

the image classifier frame by frame with the previous clustered image pixels information. Experimental results from different methods are compared. Since the image clustering with the unsupervised Bayesian on-line learning is better than other methods such as the K-means or the Maximum Likelihood based probability methods, it is used in the proposed tracking system.

- Chapter 6 presents the robust template matching and updating algorithms. After the image is segmented and clustered, the target's $2D$ region is localized continuously in the image scene. In this chapter the general SSD (correlation) based template matching method is described first. Since the matching by the conventional methods have several problems, the novel Bayesian probability fusion based robust template matching is proposed. The novel template updating strategy based on this matching method is also proposed. Experimental results are shown to address the performance improvement from the Bayesian fusion based template matching.
- After different data from the vision sensor and vehicle's inertial sensors are obtained, in Chapter 7 the target's $3D$ world coordinate velocity and position are estimated for the tracking purpose. In this Chapter the target's $3D$ position and velocity estimation procedures are developed by fusing the target's visual features with the camera motion parameters based on the velocity compensation method. Next in Chapter 7, the Extended Kalman Filter is formulated for the target dynamic state estimation. Since with only the simple linear dynamic models the tracking performance is not robust, the IMM based multiple linear dynamics model fusion is proposed to make the approximations of the target complex dynamic motion properties. Next, the $1 - step - lag$ OOSM solution is implemented with the Extended Kalman Filter to solve the out-of-sequence measurements problem encountered in the sensor fusion based tracking scheme. The IMM approach and OOSM solution improve the estimation accuracy and robustness.
- Chapter 8 presents extensive experimental results to demonstrate the tracking system's performance. The tracking system is tested under different environmental conditions and different object states. Discussions are made based on these

experimental results.

- Conclusions of the thesis are drawn in Chapter 9. A technique for “Vision based Target Tracking for Autonomous Vehicles” is proposed and demonstrated. Future improvements are suggested for the proposed visual target tracking scheme.

Chapter 2

Literature Review

2.1 Introduction

In order to make autonomous vehicles accurately sensing, detecting and tracking objects of interest, research groups around the world have been working towards advanced navigation algorithms and smart autonomous vehicles (or mobile robots).

In the first part of this Chapter, different algorithms using vision sensors for the autonomous vehicle target tracking are reviewed. Visual target tracking algorithms for the autonomous land, underwater and aerial vehicles navigation are studied and compared. The generally well-known visual target tracking algorithms are reviewed in the second part of this Chapter. The tracking ideas and methodologies from these algorithms can be introduced into the autonomous vehicle visual target tracking. The whole tracking scheme can be formulated based on these ideas. Since each tracking method has its own merits and demerits, it is necessary to have a fusion methodology to combine the advantages of each method to compensate for their disadvantages. Also autonomous vehicles are equipped with different sensors. The sensors' data are redundant and complementary. Fusing these information can lead to a more robust feature extraction and tracking. So the general approaches for data fusion are reviewed in the third part of this Chapter.

2.2 Vision based Target Tracking for Autonomous Vehicle Navigation

The targets of interest can normally be the moving cars, moving persons or any other moving objects which autonomous vehicles need to track for navigation purposes. Here the visual target tracking methods are reviewed in three different categories based on the applications in the land, underwater and aerial environments. Generally autonomous land vehicles are more widely used and more algorithms are developed for the land vehicle's navigation. So in this section the literature survey focuses more on the visual target tracking algorithms for autonomous land vehicle navigation.

2.2.1 Visual Target Tracking for Autonomous Land Vehicle Navigation

In general the applications of visual target tracking for Autonomous Land Vehicle navigation include some sorts of landmarks tracking, people following, multiple mobile robots cooperation, vehicles localization and map building, visual target tracking with pan-tilt camera platforms and so on. According to their applications, different methods are categorized and reviewed as follows. It should be pointed out that the classification in this part is not absolute because algorithms from different categories can be intergraded together to achieve the navigation goal.

2.2.1.1 Visual Landmark Tracking for Autonomous Land Vehicles Navigation

Visual landmark tracking is one kind of vision-based methods for the autonomous land vehicle navigation. Landmarks are divided into two classes: natural or artificial. Generally natural landmarks are selected in the scenes in consideration of their particular characteristics. Autonomous vehicles learn those characteristics or keep the features of the landmarks in memory and recognize them using the neural network or some matching techniques while they move [38] [148] [188]. Mosaic images of the outdoor environments

are also used for the image matching based method in [118].

On the other hand an artificial landmark is often designed with a specific pattern or color in consideration of its detection algorithm. For example, a landmark that has a bar code or a specific shape pattern such as the sine waves has been proposed. Recently, Briggs et. al. [17] use the self-similar gray pattern landmarks for the navigation and localization aids. In [81] Jang et. al. propose a simple artificial landmark model, which can be used for the self localization of indoor mobile robots. An effective visual detection and tracking algorithm for this landmark is proposed. A pair of color points neighboring each other have been used as a sample to represent the probability density in the Condensation algorithm. Under the assumption of affine cameras and only with the information of a single landmark in a single image, they have presented a localization algorithm to estimate the absolute position accurately. In [217] Wei et. al. propose a visual landmark tracking algorithm for docking unicycle-like vehicles based on the hearing-only information. An omni-directional panoramic camera is used to detect visual landmarks around the docking station and provide bearing (or heading) data for each observed landmark. In their method a robust and computationally cheap visual blob detection algorithm is proposed. The artificial landmarks used consist of two adjacent large color blobs. Using two adjacent color blobs improves the noise rejection over the single color blob extraction algorithms.

Normally, the robust extraction of natural landmarks is a difficult task. And the artificial landmark methods that use peculiar contour, color or edge information highly depend on the low-level image processing results, which are influenced by the noise and de-focus phenomenon very much. Also, artificial landmark methods are not robust under the geometrical background variations such as the rotation, camera zoom and viewing direction changes.

2.2.1.2 Human Following for Autonomous Land Vehicle Navigation

Several approaches are proposed to use visual target tracking for human following by autonomous vehicles.

In [70] Hirai et. al. present a visual tracking system for a human collaborative mobile robot. The robot tracks the human back and shoulder to follow a person. Normally it is not easy to keep tracking the human back if the background is so complex and cluttered. They solve the problem by choosing the texture of clothes and the human shoulder images as the template patterns to be detected and identified. This tracking system requires to know the special visual features of the target (the texture of the clothes and human shoulder). Such tracking systems are not suitable to track unknown objects appearing in an unknown situation.

One kind of research work on the autonomous vehicle human following can be found from Morioka et. al. [114] [145] [144]. According to Morioka et. al. [145]'s human-following work, an intelligent space (ISpace), which is an intelligent environment with many intelligent sensors, is provided. The mobile robot cooperates with multiple intelligent sensors, which are distributed in the ISpace. The distributed sensors recognize the target human and the mobile robot, and give control commands to the robot. CCD cameras are used as a kind of sensors of DINDs (Distributed Intelligent Network Devices) for ISpace. Location information of the human and mobile robots is obtained by stereo vision processing and human do not need to hold any special tags. However there is a major drawback with this kind of human-following method. The system strictly needs the ISpace. Normally for an unknown or unstructured environment, such an ISpace is not available, thus in order to achieve the visual target tracking, the vehicle's onboard sensors should mostly provide the accurate target dynamic information.

In [83] Jensen et. al. design a mobile robot, which at the same time executes a pre-programmed tour in a public exposition and allows for complex, collaborative interactions with the non-experienced visitors. In their system, they have dedicated two tasks to gather the information of the human presence in a public environment: a color camera based face tracking and a motion tracking based on the information from the laser ranger finder. The main steps of visual face detection and tracking in their system include: the skin color detection; contour extraction and filtering; tracking. Information gathered from the face tracking together with the motion tracking helps to verify the presence of the visitors.

In [153] Nishiwaki et. al. present a humanoid walking control system that generates body trajectories to follow a given desired motion on-line. They implement the system by making the robot track and follow a moving person based on the stereo vision feedback. The visual tracking consists of 3 parts: a stereo vision processing for target detection and 3D position estimation in the camera coordinates; a planning of the desired future torso movements during one step; a camera posture and gaze direction control with the self-motion compensation. While the mobile robot and human are both moving, the color segmentation and thresholds are utilized to detect the relative human's direction. Then a real-time depth map generation algorithm is employed to measure the distance to the human.

Kwon et. al. [111] present an efficient human following algorithm for a mobile robot using two independent moving cameras. In order to control the camera's pan/tilt motions, they have presented an image-based PTU control algorithm using a lookup table that stores the correspondences between the camera pan/tilt angles required to keep a target in the center of the image frame and the pixel displacements produced by the target in the image plane. The major problem with this method is that the object tracking is accomplished with a simple color histogram based algorithm. Using color information of the person's specific appearance, it calculates the centers of masses of the segmented color-blobs in each of the two images that form the conjugate pair of images in a tracking sequence. The current viewing direction of each camera in their system is adjusted so that the center of mass becomes the center of the image frame. However, a change in illumination can induce shifts in the center of mass of the blob being tracked in the two camera images, which makes the target tracking fail.

Ogata et. al. [155] propose a tracking system employing a visually controlled aerial robot which recognizes the motion of the specified person. They propose the motion recognition technique employing MHIs and eigenspaces. The human region is extracted by its color information.

In conclusion all these methods detect and track human's motion based on human's special visual features or the additional sensors in the environment. It limits the applications of such algorithms to track unexpected or different objects in an unstructured

outdoor environment.

2.2.1.3 Visual Target Tracking for Autonomous Land Vehicle Localization and Map Building

Localization and target-tracking are both challenging yet essential and have a wide range of applications in mobile robotics. Localization can be defined as determining the position of an object within a reference coordinate system, and tracking consists of constructing a trajectory given a collection of spatially and temporally coherent localizations. Several methods have been proposed using visual target tracking for the autonomous vehicle localization and map building.

In [65] Hajjawi et. al. present an algorithm for visual position tracking of individual cooperative robots within their working environment. Initially, they present a technique suitable for visual servoing of a robot towards its landmark targets. Secondly, they present an image processing technique that utilizes images from a remote surveillance camera for localization of the robots within the operational environment. In their algorithm the surveillance cameras can be either stationary or mobile. The supervisory control system keeps tracking of relative locations of individual robots and utilizes relative coordinates information of the robots to coordinate their cooperative activities.

Burschka et. al. [20] present a real-time mobile navigation system and an approach for the vision-based Simultaneous Localization and Mapping (SLAM) based on a second generation of the image processing library, XVision. They show how the multiple tracking and low-level image processing primitives for color, texture and disparity can be combined to produce vision-guided navigation systems. The applications they discuss make use of XVision capabilities to solve the temporal correspondence problem by tracking an image feature in a given image domain.

In [35] Dao et. al. present a simple linear method for localizing an indoor mobile robot based on a natural landmark model and a robust tracking algorithm. The landmark model consists of a set of three or more natural lines such as baselines, door edges and linear edges in tables or chairs to take the advantages of fast landmark detection.

The canny operator and Lucas-Kanade algorithm are used for the effectively detecting and tracking of the landmark model. Next, a quick localization method for mobile robots from correspondent lines is proposed by adopting a linear technique. Based on assumptions on indoor environments, a complex nonlinear problem for the 3D pose determination using lines is converted to an iterative linear problem, which makes it possible to apply the proposed algorithm for real-time applications. However, in their system the line detection methods are not quite accurate and robust and with some additional condition constraints.

2.2.1.4 Visual Target Tracking with Pan-Tilt Camera Platforms in Autonomous Land Vehicle

Ego-motion estimations or pan-tilt cameras' motion control are one of the key issues in autonomous vehicle navigation, especially in demand for moving objects tracking.

In [98] Karlsson et. al. develop a lightweight, robust real-time tracking system used on an experimental geo-referenced cameras platform. The purpose of their system is to study the benefits of combining image processing with navigation data that should be available from the control system of any AGV system. Their experiments show that by using a Kalman Filter the tracking algorithm can handle objects' large movements between images and it becomes more resistant to the occlusions. However, their tracking system is designed mainly based on the image brightness, which gives much less robust performance under the environmental lighting changes.

The objective of the research work in [21] is to derive the orientation of a pan-tilt camera fitting a drone in order to track a target and to maintain its position in the middle of the image. To ensure the real-time video operation, an algorithmic solution integrating a successive-step and multi-block search method is implemented, thus allowing tracking with complex target displacements. The micro-controller uses this information to manage the camera orientation. With a certain regularity in the evolution of the target model, this system is sufficiently robust to track deformable targets in real images. However their technique has limitations when the target is close to the cameras. Also only a simple linear interpolation method is carried out. The drone localization and

attitude are not considered in the algorithm. In addition, the implementation of a $2D$ visual camera control requires a priori knowledge of the $3D$ target model.

Tomono et. al. [208] present a method of planning a path on which the mobile robot with a pan-tilt camera can find the target objects under spatial uncertainties. The object recognition is normally based on feature matching between models and the image data. However there are several problems in this method. For example, in the case that the salient features to identify the target object may center on their particular faces, the robot has to move around the object to find the features. If the possible locations of the target object are in a wide area, the robot has to move around the area to search the object. This paper addresses these problems by a probabilistic approach. Given an initial roughly-planned path, the proposed method optimizes it with respect to the travel time, high pass-ability, and a high probability of finding the target. The method defines a path evaluation function based on these factors and finds a suboptimal path by solving the nonlinear optimization problem of the path evaluation function. Their method combines the visibility constraint and the conventional constraints of travel time and collision avoidance for the mobile robot navigation.

Zhang et. al. [230] develop a pan-tilt visual tracking system to dynamically track moving targets using vision-based control. The algorithm includes the color-based segmentation, data pre-processing and active parameters adjustment. Since the computationally expensive techniques are inapplicable, they focus on the use of specific color properties to identify the objects of interest.

2.2.1.5 Visual Target Tracking for Multiple Mobile Robots Cooperation

Multiple mobile robots (autonomous vehicles) cooperation means that each mobile robot plans its path based on other robots' navigation information. The robots cooperate with each other to complete the navigation tasks. Several methods based on the visual target tracking have been proposed in this area.

In [228] a real-time visual tracking algorithm for MRFS (Multiple Robot Fishes cooperation System) is described. They give a description of the operation process for the

vision subsystem, and propose an adaptive segmentation method based on the color information. Color information is the foundation of their object identification. In MRFS, halls, obstacles and robot fishes are equipped with specified colors properties.

In [135] the problem of estimating and tracking the motion of a moving target by a team of mobile robots is studied. Each robot is assumed to have a directional sensor with limited ranges, thus more than one robot (sensor) is needed for solving the problem. A sensor fusion scheme based on the inter-robot communication is proposed in order to obtain the accurate real-time information of the target's position and motion. Accordingly a hierarchical control scheme is applied, in which a consecutive set of desired formations are planned through a discrete model and low-level continuous controls are executed to track the resulting references.

Betser et. al. [14] describe a tracking system relying on active contours for the target exaction and an Extended Kalman Filter for the relative pose estimation. Their work represents the first step towards treating the general problem for the control of several unmanned autonomous vehicles flying in formulation using visual information. The Extended Kalman Filter is improved by introducing additional image information available to the vehicle with a fixed forward-pointing monocular camera. Active contours are used to track the follower in the image plain and provide the Kalman Filter with the required input. There are several drawbacks in this kind of methods. They ignore the additional angle measurements found in the equivalent $3D$ range estimations and also they do not consider the target's acceleration effects.

In [150] Ng et. al. have formulated an algorithm that has coordinated the movements of multiple robots to follow a search tactic collectively in an unknown and cluttered environment. First they develop the individual robot reactive behavior that make their coordinated movement possible. Their algorithm requires every robot to be programmed with the same set of primitive behaviors: (1) obstacles negotiation; (2) homing; (3) flocking and (4) searching, with obstacles negotiation being the most important and searching being the least important. According to different environment stimulants, the robots adopt one of these behaviors at a time according to their order of importance for the cooperation purposes.

2.2.1.6 Other Visual Target Tracking Methods for Autonomous Land Vehicle Navigation

Traditional visual target tracking techniques usually ignore the presence of obstacles and focus on imaging and target recognition issues. The papers [60] [113] introduce a new visual tracking algorithm for autonomous vehicle navigation when the target moves unpredictably and no prior map of the environment exists. Their algorithm computes a motion strategy based exclusively on the current sensor information, while no global map or historical sensor data is required. The algorithm is based on the notion of escape risk and the computation of an escape-path tree. Their proposed algorithm governs the motion of autonomous vehicles based on current measurements of the target's position and the location of the local obstacles. Their approach is combinatorial in the sense that the algorithm explicitly computes a description of the geometric arrangement between the target and observer's visibility region produced by the local obstacles. The algorithm computes a continuous control law based on this description. However most failures of this system are due to the shortcomings of their simple visual target detection algorithms. Their method is also limited to a $2D$ workspace without considering the problems in the $3D$ space.

Tracking targets robustly by vision is very difficult for autonomous vehicles running on irregular terrain in the natural environment, because the image deformation caused by rolling and pitching of the camera, as well as the relative movement between the target and camera, affect the tracking ability greatly. One approach to cope with such problems is matching the target image with many affine transformed candidate images while tracking. But when the number of candidate images gets larger, such an approach is not available to the real-time tasks due to the computational cost. In [44] Ding et. al. propose a new Robustness Analysis for Tracking (RAT) to improve the tracking ability. RAT is the analysis based on features of the object image, where three parameters: 'Detectability', 'Robustness for Depth (RBD)' and 'Robustness for Rotation (RBR)' are defined. Much more robust templates can be found by analyzing the object image using RAT before the tracking task is performed.

In [19] Burschka et. al. present an approach for scene classifications in dense disparity

maps from a binocular stereo map. The classification result is used for the indoor mobile robot tracking and navigation purposes. The 3D model of the scene is also derived directly from the disparity image. The classification of the scene helps to decide, which objects are interesting and should be monitored as well as which behavior is appropriate depending on the current structure. The robot can switch from the wall following in hallway environments to localization based on the corner structures, etc. However, the shortcoming of this system is that the applied algorithms used for the dynamical composition of tracking primitives are highly dependent on the current environmental structures.

In [142] Mori et. al. propose a ball tracking and catching strategy called GAG (short for “Gaining Angel of Gaze”) that enables a mobile robot to track and catch a ball flying in the three-dimensional space. The mobile robot receives a visual feedback control scheme based on GAG and then the proposed scheme enables the robot to track and catch a ball flying in the three-dimensional space by using a monocular vision system.

In [229] Yu et. al. propose a correspondence based method, which applies the Iterative Closest Point (ICP) algorithm to match feature points on ground plain. Since the outliers in the scene contribute false measurements for estimation, they introduce a stereo vision-based method to detect free-space on the road plain. They extract the edge points in the free-space as primitives, which avoid the limit of the rigid scene hypothesis.

Other visual target tracking methods for the autonomous land vehicle navigation are: The paper [147] deals with the problem of computing the motions of a robot observer in order to maintain the visibility of a moving target; Taking inspiration from the visual system of the fly, the paper [69] describes and characterizes a monolithic analog very large-scale integration sensor, which produces control signals appropriate for the guidance of an autonomous robot to visually track a small moving target; The visual vehicles following systems are also presented in [34] [201].

Generally all the above methods have several common drawbacks: they normally do not take into account the complex 3D motions of the cameras; they do not have appropriate target dynamic models for the tracking estimation; the target visual features excitations from the images are based on simple low-level image processing methods or the target’s

special visual features, which are less accurate and robust; they can not track the target's 3D dynamics in the world coordinate.

A common problem in all the land navigation systems is caused by the environmental shadows, changing illumination conditions, changing colors, etc. This can impose serious limitations on the performance of a navigation system.

2.2.2 Visual Target Tracking for Autonomous Underwater Vehicle

In the underwater environment visual target tracking can be broadly used for the Autonomous Underwater Vehicle (AUV) navigation. For example in order to study the behavioral patterns of unknown underwater life forms, it is necessary to observe them carefully for a longer period. It is therefore, important to stay close to the object being observed by moving with it such as the fish following. Also, for a reliable functioning of underwater man-made systems, it is necessary to maintain them by routine observations. Therefore, the underwater exploration and maintenance require proper observations and, as a result the close navigation along or with these objects is necessary [222]. The visual target tracking system can be used for these autonomous underwater vehicle navigation applications.

Underwater pipe inspection is one example of a class of problems that bear many similarities with visual target tracking for autonomous land vehicles. Extracting the contours of a pipe is equivalent to extract and track the landmarks. In [174] Rives et. al. take a visual servoing approach and devise a controller that take the inputs as the lines extracted from the image of a pipe and uses this information to generate steering commands for the ROV Vortex vehicle. In [4] Arjuna and Tamaki propose a sensor fusion technique for the Autonomous Underwater Vehicle to track underwater cables. They propose a sensor fusion scheme using the dead reckoning position uncertainty with a 2D position model of the cable to predict the region of interest in the image. They solve two practical problems encountered in the optical vision based systems in underwater environments: first, the navigation of AUV when the cable is invisible in the image; second, the selection of

the correct cable (interest feature) when there are many similar features appearing in the image.

Arjuna et. al. [5] also propose the vision based tracking system for the underwater docking using correlation based underwater image template matching.

One of the most recent underwater visual target tracking methods is proposed in [222] for AUV navigation. In Yang's work the objects of interest are extracted from the images by using the dynamic properties (optical flow techniques) and their optical features (color, texture, shape and so on). The consecutive dynamic behavior of the objects of interest is then estimated based on the current dynamics. By using this predicted dynamics, the amount of data for the region of interest identification can be reduced. This also increases the speed of processing for the hardware available in the small and limited hulls of AUV. After the image dynamics and feature position information are fused with the AUV's other onboard sensor data, the navigation commands are derived for AUV to track the object.

2.2.3 Visual Target Tracking for Autonomous Aerial Vehicle

Development of the Autonomous Aerial Vehicle (AAV) (such as the unmanned helicopters) has been an active area of research for several years. AAVs have been used as test beds to investigate problems ranging from control, navigation, and path planing to object tracking and following [178].

An early approach to the AAV landing [139] decouples the landing problem from vision-based target tracking. Nowadays several techniques have been implemented for vision based landing of an AAV on stationary or moving targets [139]. The problem of landing as such is inherently difficult because of the instability of the AAV near the ground [189]. Also, since the dynamics of a helicopter is nonlinear, only an approximate model of the helicopter can be constructed [178].

In [182] a vision-based target tracking approach to AAV landing is presented. Their landing pad has a unique shape which makes the problem of identification of the landing pad much simpler. In [57], a vision-based solution is given for the safe landing-site

detection in the unstructured terrain, where the key problem is for the onboard vision system to detect a suitable place to land without the aid of a structured landmark, such as a helipad. The University of California, Berkeley (UC Berkeley) team has proposed a real-time computer vision system for tracking a landing target [185] [189] and have successfully coupled it with a helicopter controller to achieve landing [183].

In [177] Saripalli et. al. have presented the design and implementation of a real-time vision-based system for detecting a landing target (stationary or in intermittent motions) and a controller to autonomously land the AAV on the target. Their method relies on the assumptions that the landing target has a well-defined geometric shape, and all the feature points of the landing target are coplanar. In [177] they use invariant moment descriptors for detecting and landing the AAV on the target. They do not impose any restriction on the shape of the landing pad except that it is planar. However there exists the problem of safely and precisely landing the AAV in the unstructured harsh 3D environment.

The visual target tracking approach from [178] differs from the prior approaches in two ways. First, they impose no constraints on the design of the landing pad except that it should lie on a two-dimensional plain. Secondly, they use moment descriptors to determine the location and orientation of the landing target. Their algorithm is not only able to detect and land on a given target, but also able to track a target which moves intermittently, and land on it.

In [179] Saripalli et. al. present a vision-based algorithm designed to enable the AAV to land on a moving target. The AAV is required to identify a target, track it, and land on it while the target is also in motion. They use Hu's moments of inertia for the precise target recognition and a Kalman filter for target tracking. Based on the output of the tracker a simple trajectory controller is implemented which (within the given constraints) ensures that the AAV is able to land on the target.

In [196] it has been shown that Unmanned Air Vehicles equipped with a 2-degree-of-freedom pan-tilt camera can be used for the long-term autonomous observation of stationary ground targets. The environmental aspects such as wind and sunlight have an influence on the quality of the images taken by the camera. These influences can set

limits to the time span of continuous target observation, especially when the range of camera motion is mechanically limited. Two different camera limit specifications with the realistic applicability have been analyzed for these potential problems. Solutions to these problems have been found by applying circle-based flight maneuvers including commanded sideslip for heading adjustment.

In conclusion several limitations exist with the above visual target tracking algorithms for AAVs: Some of their estimators can only track the target in a single dimension; Some of them can not track the targets in all the six degrees of freedom; They all use the algorithm based on the intensity of the image for object detection; Some of them assume that the object is planar, which is quite restrictive in nature. A better object detector and a camera calibration routine can be integrated with these algorithms so that the coordinates of the tracked object obtained during the vision processing stage are much more accurate; These algorithms can only track single object and can not pursue and land on evasive targets.

2.3 Visual Target Tracking Technologies

In recent times a vast number of algorithms have been proposed in the field of visual target tracking. According to the recent literature review paper of [73], the prerequisites for effectively tracking moving target using CCD cameras generally include the following stages: detection of motion and target tracking.

2.3.1 Motion Detection

Nearly every visual target tracking system starts with motion detection. Motion detection aims at segmenting the regions corresponding to the moving objects from the rest of an image. Subsequent processes of target tracking are greatly dependent on it. The process of motion detection usually involves environment modelling, motion segmentation, and object classification, which intersect each other during processing.

- The active construction and updating of environmental models are indispensable

to target tracking. Environmental models can be divided into $2D$ models in the image plain and $3D$ models in the real world coordinates. Due to their simplicity, $2D$ models have more applications. $2D$ models normally include the environment modelling for the fixed cameras [55] [104] [172] [199]; the pure translation (PT) cameras [186]; the mobile cameras [207]. Regarding the $3D$ environmental models [232], the current work is still limited to the indoor scenes because of the difficulty of $3D$ reconstructions of outdoor scenes.

Autonomous vehicles are working in an unknown or unstructured $3D$ situation with the continuous environmental changes. In such a situation it is quite difficult to model the environment continuously and accurately. Therefore the environment modelling is not suitable for motion detection for autonomous vehicle visual target tracking.

- Motion segmentation in image sequences aims at detecting the regions corresponding to the moving objects such as the vehicles and human. Detecting moving regions provides a focus of attention for the later processes such as tracking and estimation because only these regions need to be considered in the later processes. At present, most segmentation methods use either the temporal or spatial information in the image sequence. Several conventional approaches for motion segmentation are temporal differencing [123] [210]; background subtraction [68] [136] [195]; optical flow vectors [137] [138] and so on [29] [55] [198].

Both the background subtraction and temporal filtration methods offer the solutions to the problem of motion based image segmentation but they do have a number of drawbacks. Neither of these methods succeed in an environment where there is a dynamically changing background. Examples of such situations could include when the camera itself is moving or where areas of non-salient motions are apparent. Background subtraction has the added limitation of requiring a representation of a scene absent of moving objects. This representation requires either the prior knowledge of the environment or the construction of such a background scene based on the average of possibly hundreds of motion filled the frame shots.

Optical flow technologies do not have the above mentioned problems. Optical flow methods can work in an unknown or changing background and it does not require

a prior information about the environment or the object. However optical flow methods may require more computational time.

- Different moving regions may correspond to different moving targets in the natural scenes. For instance, the image sequences captured by the moving cameras mounted on the autonomous vehicle probably include humans, vehicles and other moving objects. To further track objects and analyze their behaviors, it is essential to correctly classify moving objects. Object classification can be considered as a standard pattern recognition issue. At present, there are two main categories of approaches for classifying moving objects: motion-based classification [33] [122] and shape-based classification [29] [123] [110]. These two common approaches can also be effectively combined for the classification of moving objects [194].

Shape based classification methods always require to have the priori knowledge of the objects. However motion based classification methods are usually independent of the object's other visual features.

After motion detection the target tracking systems track moving objects from one frame to another through an image sequence. The target tracking algorithms are reviewed next.

2.3.2 Target Tracking

The visual target tracking algorithms usually have considerable intersections with the motion detection during image processing. Tracking over time typically involves matching objects in consecutive frames using features such as points, lines or blobs. Useful mathematical tools for tracking include the Kalman filter, the Condensation algorithm, the dynamic Bayesian network, and the geodesic method, etc.

Tracking methods can be commonly divided into several major categories: region-based tracking, active-contour-based tracking, feature-based tracking, model-based tracking, template-matching-based methods and so on. It should be pointed out that this classification is not absolute in that algorithms from different categories can be integrated together [82].

- Region-based tracking algorithms track objects according to the variations of the image regions corresponding to moving objects. For these algorithms, the background image is maintained dynamically [102], and the motion regions are usually detected by subtracting the background from the current image.

Although these methods work well in scenes containing only a few objects (such as highways) [136] [218], region-based tracking algorithms cannot reliably handle occlusions between objects. Furthermore, as these algorithms only obtain the tracking results at the region level and are essentially procedures for the motion detection, the outline or 3D dynamics of objects cannot be acquired. (The 3D dynamics of an object consists of the position and velocity of the object). Accordingly, these algorithms cannot satisfy the requirements for the autonomous vehicle target tracking against a cluttered moving background or with multiple moving objects.

- Active contour-based tracking algorithms track objects by representing their outlines as bounding contours and updating these contours dynamically in successive frames [11] [56] [141] [219]. These algorithms aim at directly extracting shapes of subjects and providing more effective descriptions of objects than region-based algorithms [107] [159] [164]. In contrast to the region-based tracking algorithms, active contour-based algorithms describe objects more simply and more effectively and reduce the computational complexity [77] [130] [234]. Even under the disturbance or partial occlusion, these algorithms may track objects continuously. Another advantage of the above mentioned system is that it inherently contains information of a global structural nature which makes it less sensitive to weak or missing image features.

However contour-based tracking systems suffer from the limitations of their segmentation techniques. The tracking precision is limited at the contour level. The recovery of the 3D dynamics of an object from its contour on the image plane is a demanding problem. A further difficulty is that the active contour-based algorithms are highly sensitive to the initialization of tracking, making it difficult to start tracking automatically. Another problem with the purely contour-based systems is that they neglect the information contained within the body of an ob-

ject [76], focusing purely on its external area. In the deformable contour methods once an area is chosen (manually) for tracking, the template is deformed via attractions to areas of the high intensity gradients (edges), motion boundaries and color/grayscale similarity in the future frames. Another limitation is that due to the temporal based segmentation a fixed background environment is often required.

- Feature-based tracking algorithms perform recognition and tracking of objects by extracting elements, clustering them into higher level features and then matching the features between images. Feature-based tracking algorithms can further be classified into three subcategories according to the nature of selected features: global feature-based algorithms [160] [166] [180]; local feature-based algorithms [28] [130]; and dependence-graph-based algorithms [49]. The above three methods can also be combined in [80]. In general, as they operate on $2D$ image plains, feature-based tracking algorithms can adapt successfully and rapidly to allow real-time processing and tracking of multiple objects which are required in heavy thruway scenes, etc. Feature-based tracking algorithms can handle partial occlusion by using information on the object motion, local features and dependence graphs.

However, there are also several serious deficiencies in the feature-based tracking algorithms. The recognition rate of the objects based on $2D$ image features is low, because of the nonlinear distortion during the perspective projection and the image variations with the viewpoint's movement. These algorithms are generally unable to recover the $3D$ dynamics of objects. The stability of dealing effectively with occlusion, overlapping and interference of unrelated structures is generally poor. Some feature-based target tracking methods only track objects containing the specific visual features such as the skin colors. However skin-color-like areas in the background may be confused with the real regions of interest in such a situation.

As the visual target tracking system for autonomous vehicles works in an unknown or unstructured situation, the target's specific features are generally unavailable or unpredictable. Also the environment conditions are changing, the visual features may also change. Thus it is not suitable to develop a tracking system mainly based on the target special visual features.

- Template matching based systems use the algorithms that allow a measure of coherence between templates and their possible matches in the images. It is the goal of such matching systems to identify the features of objects of interest and localize it in the images [75] [76].

The underlying assumption behind template matching is that the appearance of the object remains the same throughout the entire image sequence. This assumption is generally reasonable for a certain period of time, but eventually the template is no longer an accurate model of the appearance of the object (appearance changes or occlusion problems). If without an priori object appearance model, the target's appearance model has to be updated during tracking. A “naive” solution to this problem is to update the template every frame (or every t frames) with a new template extracted from the current image at the current location of the template [133] [187]. The problem with this “naive” algorithm is that the template may drift or have some appearance changes. The “naive” updating will also fail at the presence of occlusions or abrupt changes in the lighting condition [133]. Another major limitation of template matching methods is their great computational time. Also to implement a template based tracking system it is always necessary to perform some form of preprocessing or manual user interaction to acquire the region of interest for tracking. Often the image segmentation is based on background subtraction, inherently involving the limitations mentioned earlier.

- Model-based tracking algorithms track objects by matching projected object models, produced with prior knowledge, to image data. The models are usually constructed off-line with the manual measurements, CAD tools or computer vision techniques. As the model-based rigid object tracking and model-based nonrigid object tracking are quite different and also normally there are usually two kinds of objects of interest tracked by autonomous vehicles, the model-based human body tracking (nonrigid object tracking) and model-based vehicle tracking (rigid object tracking) are reviewed separately.

As to the model-based vehicle tracking, 3D wire-frame vehicle models are mainly used [58] [162]. The research groups at the University of Reading [201] [202] [203] [204], the National Laboratory of Pattern Recognition (NLPR) [126] [225] and the

University of Karlsruhe [64] [106] [108] have made important contributes to 3D model-based vehicle localization and tracking.

The general approach for the model-based human body tracking is with the following steps. First, the pose of the model for the next frame is predicted according to the prior knowledge and tracking history. Then, the predicted model is synthesized and projected into the image plain for the comparison with the image data. A specific pose evaluation function is needed to measure the similarity between the projected model and the image data. According to different search strategies, this is done either recursively or using sampling techniques until the correct pose is finally found and is used to update the model. Pose estimation in the first frame needs to be handled specially. Generally, the model-based human body tracking involves three main issues: construction of human body models [1] [36] [37] [93] [97] [154] [165] [175] [192] [212]; representation of prior knowledge of motion models and motion constraints [16] [26] [152] [187] [233]; prediction and search strategies [12] [37] [78] [79] [72] [127] [212].

Compared with other tracking methods, model-based tracking algorithms have the following main advantages. By making use of the prior knowledge of the 3D contours or surfaces of objects, the algorithms are intrinsically robust. The algorithms can obtain better results even under occlusions (including the self-occlusion for humans) or interference by nearby image motions. As far as the model-based human body tracking is concerned, the structure of human body, the constraint of human motion, and other prior knowledge can be fused. As far as the 3D model-based tracking is concerned, after setting up the geometric correspondence between 2D image coordinates and 3D world coordinates by the camera calibration, the algorithms naturally acquire the 3D pose of objects. The 3D model-based tracking algorithms can be applied even when objects greatly change their orientations during the motion.

Ineluctably, the model-based tracking algorithms have the disadvantages such as the necessity of constructing the models, high computational cost, etc. Normally a prior model of the object is not quite available for autonomous vehicle target tracking in the unknown environment.

2.3.3 Optical Flow Vectors Based Target Tracking Systems

Optical flow based tracking systems are mainly using optical flow motion vectors for the target identification and tracking. Optical flow vectors estimate the motion of the image gradients within an image sequence. The ultimate goal of this estimation is then to interpolate the 3D motion of an object.

Okada et. al. [157] propose a system where the generalized gradient estimation method is used to extract optical flow vectors. From the optical flow vectors field the object's motion localization can then be achieved. The motion field is then interpreted with the knowledge of the object's shape such as human to predict and track a moving object's position. By the use of an iterative optical flow algorithm and the parallel DSP hardware, Okada et. al. [157] implement a single target tracking system in real time. Further incorporations of an active camera head also allow tracking over a wider field of view.

While several tracking methods have been proposed, tracking using a single cue such as the optical flow vectors [220], depth [30], or edge [75], may fail when the target cannot be exactly identified based on the employed cues. By integrating various cues it is possible to increase the reliability of target tracking. Okada et. al. [157] integrate optical flow and depth to extract the target region, and Yamane et. al. [221] integrate optical flow and uniform intensity regions. In these methods the target that cannot be correctly extracted from one cue can be tracked using other cues. In Yamane's tracking system [221] in order to overcome the problems associated with optical flow (or lack of) in the homogeneous regions, a method of integration is designed such that optical flow and spatial segmentation (similar to [210]) work in union. By integrating various cues it is shown that a more reliable object tracking system could be designed and that due to an increase in the information of the object, the extension could also be made to track multiple objects.

Since all the above methods assume that optical flow vectors in the target region are almost uniform, implying that the target translates almost parallel to the image plane, tracking may fail when the target moves with the general 3D motion. In 3D tracking the translation along the optical axis is less sensitive to optical flow than the translations

parallel to the image plain. Also the rotation upon an axis perpendicular to the optical axis generates the flow vectors similar to the ones generated from the translation parallel to the image plain. Therefore it is still difficult to reliably estimate every component of the general 3D motion from optical flow alone [143]. On the other hand the translation along the optical axis can be directly estimated from the depth. The rotation angle upon an axis perpendicular to the optical axis could also be computed if the resolutions of the depth map are high enough and the 3D target shapes are known. Although such a depth map with sufficient resolutions cannot be obtained by conventional methods, since the change of this rotation angle causes the shape change of the target region in the image, it can be an effective cue for estimating this rotation. Thus in order to solve the above 3D tracking problems, Okada et. al. [158] propose a method to extract the reliable target region by integrating optical flow and depth and then estimate the target state (3D position, posture and motion) using the shape of the target region, optical flow and depth. Although none of them alone can estimate the 3D target state reliably, they compensate for each other to make the better tracking performance.

One example of the optical flow based visual target tracking system used in autonomous vehicle navigation can be referred to Doi et. al. [45]. They propose a real-time navigation system which observes the human behavior and reacts to those actions. The system detects body parts as the moving areas, and a face region or region specific to human is then extracted in the detected area based on the skin color or the cloth color of the human.

Optical flow vectors can also be used to implement the idea of motion compensation for visual target tracking. In [23] Chang et. al. propose an image stabilization system using optical flow techniques. A computational scheme that facilitates the local motion vector field to estimate the global disturbing translational and rotational motions is derived. In addition to suppressing the translational disturbance, they develop the DIS system which is also capable of removing the rotational disturbance. From [23] they integrate the image velocity optical flow vectors (local motion) and the camera motions (global motion) together for the motion compensation.

However there are also some drawbacks in the previous optical flow based visual target

tracking systems. First the computational time required for the optical flow estimation is quite large. Although nowadays the computer hardware is more powerful, this is still a major problem in the real-time performance. Secondly, many optical flow based tracking systems use some previous target models (such as the human models) [158] or special visual features [45] for tracking. They can only track certain objects. The tracking estimation models used in these systems are complex and the estimation takes quite a lot of time and also is not accurate. Thirdly, the target identification procedures from the optical flow vectors often rely on the simple low-level image processing algorithms, which are less accurate and robust.

2.4 Multiple Sensor Data Fusion

Since each tracking method has its own merits and demerits, it is necessary to have a fusion methodology to combine their advantages and also compensate for their disadvantages. Also autonomous vehicles are equipped with different sensors (such as inertial sensor, camera, laser scan, sonar, radar and so on). The sensors data are redundant and complementary. Fusing these information can lead to a robust feature extraction and target tracking performance. Therefore in this part the general methods of sensor data fusion are reviewed.

The potential advantages of data fusion and integration are concluded by [129] as follows:

- The integration or fusion of redundant information can reduce the overall uncertainty and thus serve to increase the accuracy with which the features are perceived by the system. Multiple sensors providing redundant information can also serve to increase the reliability in the case of sensor errors or failures.
- Complementary information from multiple sensors allows the features in the environment to be perceived that are impossible to be perceived using just the information from each individual sensor operating separately.
- More timely information may be provided by multiple sensors due to either the actual speed of operation of each sensor, or the processing parallelism that may

Estimation methods	Non-recursive: *Weighted Average *Least Squares Recursive: *Kalman Filter *Extended Kalman Filtering
Classification methods	*Parametric Templates *Cluster Analysis *Learning Vector Quantization (LVQ) *K-means Clustering *Kohonen Feature Map *ART, ARTMAP, Fuzzy-ART Network
Inference methods	*Bayesian Inference *Dempster-Shafer Method *Generalized Evidence Processing
Artificial intelligence methods	*Expert System *Adaptive Neural Network *Fuzzy Logic

Table 2.1: The Classification of Data Fusion Algorithms from [129]

be possible to be achieved as part of the integration process [129].

2.4.1 Data Fusion Methodologies

In Table.2.1 Luo et. al. [129] review the general data fusion methodologies. Data fusion algorithms can be broadly classified as follows: estimation methods, classification methods, inference methods, and artificial intelligence methods (Table.2.1).

- **Estimation Methods.** One of the simplest and most intuitive general methods of data fusion is to take a weighted average of the redundant information provided by a group of sensors and use it as the fused value. While this method allows for the real-time processing of dynamic low-level data, Kalman Filter is predominantly preferred because it provides a method that is nearly equal in processing requirements and, in contrast to a weighted average, results in the estimates for the fused data that are optimal in a statistical sense [129]. The Kalman Filter uses the statistical characteristics of the measurement model to determine estimates recursively for the fused data [88].

- **Classification Methods.** The multidimensional features space can be partitioned into distinct regions, each representing an identification or identity class [67]. The location of a feature vector is compared to the pre-specified locations in the feature space. A similarity measure must be computed, and each observation is compared to a priori class. The multi-dimensional features space may be normally partitioned by the geometrical or statistical boundaries.

The cluster analysis methods try to establish the geometrical relationships on a set of sample data in a training process. Cluster analysis methods include the hierarchical agglomerative, hierarchical divisive, iterative partitioning methods, etc [129]. The cluster analysis is a powerful tool to classify multi-sensor data. The data scaling, selection of similarity metric, and clustering algorithms will affect the resulting clusters.

Unsupervised or self-organized learning algorithms such as the learning vector quantization (LVQ), K-means clustering, Kohonen feature map can be used for classification [105]. The K-means clustering algorithm is one of the commonly used unsupervised learning algorithms [89]. An adaptive K-means updating rule forms the basis of the Kohonen feature Map. There are also the ART, ARTMAP, and Fuzzy ART networks which do sensor fusion in the adaptive manner and automatically adjust the granularity of the classifier and are stable against the category proliferation in the presence of drifting inputs and changing environments.

- **Inference Methods.** The Bayesian inference allows multi-sensor information to be combined according to the rules of the probability theory [46]. Bayesian formulas provide a relationship between the a priori probability of a hypothesis, the conditional probability of an observation given a hypothesis, and the a posteriori probability of the hypothesis [66]. The Bayesian inference updates the probabilities of the alternative hypotheses, based on the observational evidence. New information is used to update the a priori probability of the hypothesis. Dempster-Shafter evidential reasoning is an extension to the Bayesian approach [206].
- **Artificial Intelligence Methods.** High-level inferences require human reasoning such as pattern recognition, planning, induction, deduction, and learning. A

large number of expert systems have been developed for the industrial and military applications. The inference process utilized by the expert system begins with an initial data set and the rule-based knowledge base comprising rules, frames, scripts, or semantic nets [66]. A neural network [129] consists of layers of processing elements that may be interconnected in a variety of ways. Neurons can be trained to represent sensor information and, through associate recall, complex combinations of the neurons can be activated in response to the different sensory stimuli. Fuzzy logic, a type of the multiple-valued logic, allows the uncertainty in multi-sensor fusion to be directly represented in the inference process by allowing each proposition, as well as the actual implication operator, to be assigned a real number from 0.0 to 1.0 to indicate its degree of truth [61] [197].

In conclusion data fusion can be performed at different levels: the signal level, the pixel level, the feature level and the symbol level. Appropriate fusion algorithms can be accordingly developed to fuse the data from multiple sources at different levels. In general, estimation methods have been successfully used for the signal-level sensor fusion. Classification methods can be used to extract features and fuse data at the pixel and the feature level. Inference methods can be effective for the symbol-level sensor fusion due to their capabilities of evidential reasoning. As for the artificial intelligence methods, they can be regarded as the advanced versions of the estimation, the classification, and the inference methods. Artificial intelligence methods can be model-free, rather than model-specific, and have sufficient degrees of freedom to fit complex nonlinear relationships, with the necessary precautions to properly generalize. As a result, artificial intelligence methods can effectively conduct data fusion at different levels.

2.4.2 Applications of Data Fusion for Autonomous Vehicles

Redundant and complementary sensor data can be fused and integrated using the multi-sensor fusion techniques to enhance the system capability and reliability. In recent years, benefits of multi-sensor fusion have motivated the research in the autonomous vehicle (mobile robotics) areas.

Mobile robotics presents one of the most important application areas for the multi-sensor data fusion and integration [96] [129]. When operating in the uncertain or unknown dynamic environments, integrating and fusing data from multiple sensors enables the mobile robots to achieve quick perceptions for the navigation. In conclusion the data fusion and integration of the vision, tactile, thermal, range, laser radar, and forwards looking infrared sensors play a very important role for the autonomous vehicle systems.

2.4.2.1 Fusion based Visual Target Tracking for Autonomous Vehicle Navigation

Multiple sensors data fusion or multiple algorithms fusion have been widely used in the visual target tracking for autonomous vehicle navigation.

In the article of [27] a multi-sensor fusion approach between an omnidirectional vision system and a panoramic range finder is presented to dynamically localize a mobile robot. These two sensors provide some complementary and redundant data which enable to construct a robust sensorial model which integrates an important number of significant primitives. Based on this model, they treat the problem of maintaining a matching and propagating uncertainties on each matched primitive. In details their localization method is based on the tracking of significant landmarks of the environment and integrating a multi-level uncertainty propagation stage based on the usage of the Dempster-Shafer theory. For this purpose, they have developed a robust multi-criteria method which solves two problems linked to the tracking: the propagation of an uncertainty concerning the landmark tracks and the treatment of the apparition and momentary disappearance of a track.

In [42] Derrouich et. al. present a new hybrid approach for range estimation that combines inertial and visual based technologies, which allows them to calculate the image-space distance between the mobile robot and the edge lines of the 3D environment. Two frames obtained from the moving CCD video cameras and the outputs of the inertial tracking system which report the relative changes of orientations and accelerations between the two frames are integrated to estimate the image-space distance of different 3D points.

Using the Cellular Nonlinear Network (CNN) tracking method, with a combination of inertial and visual tracking technologies, Derrouich et. al. [41] have presented a direct trigonometric range estimation method from a stream of images in the static scene for the visual target tracking.

Marin-Hernandez et. al. [131] describe how to integrate tracking functions on a mobile robot in different situations: landmark tracking to guarantee the real-time robot localization, target tracking for a sensor based motion or obstacle tracking to control an obstacle avoidance procedure. These objects such as landmarks, targets, and obstacles have different characteristics. Moreover, normal tracking methods are not robust enough to work under varying environmental conditions. Tracking often fails when illumination undergoes significant changes. Some other situations where tracking methods fail are due to the cluttered background, changes in the model pose, occlusions of the target image or motion discontinuities in the object dynamics. Thus under the real-world environmental conditions, a single tracking method cannot deal with all the different tasks and situations presented in it. Several methods must be integrated on the mobile robot. In their paper a tracker controller is in charge to activate a specific method with respect to the subject type and a transition model used to recover the tracking failures. This method is validated using four trackers based on: template differences, set of points, lines and snakes. This method requests the tracker to switch, either if a target is lost, by one quick but not so robust method, and a more robust method is activated to find again the target using local image measurements, or if a target switch is necessary, for example when the robot must turn around an edge after a corridor following execution. In such a way every target is associated with the more adapted tracker and with a recovery procedure in case of failures.

In the paper of [47] a multiple-sensor multiple-target tracking (MS-MTT) approach for the Autotaxi system is proposed. The system consists of two basic components: the sensor-level tracking and multiple-sensor track fusion center. Each sensor in the sensor level is considered as an intelligent one which generates its own track files. Thus, the task of the fusion center is to fuse the local track files to produce a more accurate and reliable single system track file. This is performed in three stages: data alignment, track-to-track association and track fusion. In their paper a decentralized sequential

data association and fusion structure, referred to as the SMNDNN with MDWOR logic method, has been considered for the Autotaxi application.

Another example of the implementation of sensor fusion algorithms for the mobile robot target tracking is proposed by [129]. The experimental setup of their system consists of one autonomous mobile robot and one multi-sensor based electrical wheelchair. The mobile robot (Nomad 200 platform) is a three-wheel mobile platform equipped with a vertical sliding manipulating arm and other sensory modules. The experimental target is the multi-sensor based electrical wheelchair which is developed by them. Their system contains two major agents for the local decisions. One is the target-tracking agent whose inputs are the target position measurements from the fusion of the ultrasonic and vision sensors and the other is the collision-avoidance agent whose inputs are the surrounding range measurements from the fusion of the 16 ultrasonic sensors. The final decision calculated by fusion of these two local decisions is the absolute driving velocity of the mobile robot.

2.5 Summary

The focus of this thesis is mainly on the vision based autonomous vehicle navigation. The challenge is that normally in an unstructured environment there are no regular properties that could be perceived and tracked for navigation (such as landmarks). In such cases the vision system may only make use of at most a generic characterization of the possible obstacles in the environment as described in [43]. The obstacles's 3D dynamic information together with its visual features are then tracked for navigation. Therefore visual target tracking algorithms can be an alternative better solution than the other navigation methods.

Although numerous researchers have made efforts to develop visual target tracking systems for autonomous vehicle navigation, there are still several remaining research challenges.

First, as the visual tracking system designed for autonomous vehicles works mainly in an unknown environment, the target's specific features are generally unavailable or

unpredictable. Also as the cameras are moving with the vehicle, the visual features of the environment may also change continuously. For example as the light attenuates exponentially with the distance in the air in an outdoor environment, it makes the quality of the camera images poor. In such a situation the color of the object of interest will be attenuated and the normal tracking algorithms will meet great difficulties. Also there may be many unknown colorful objects in the same environment, which can be misinterpreted for the object of interest. So it is not suitable to develop a tracking system mainly based on the target's special visual features or models such as the color or shape.

Secondly, in the normal visual target tracking systems the target's relative dynamics to the cameras is estimated for navigation purposes. This relative information is obtained from the vehicle's onboard sensors without considering the vehicle's self movement. The relative information is not available for other applications in the same environment. Also the relative information is less accurate due to the vehicle's self rotational and translational motions, which are normally ignored in the estimation processes. Instead of using the relative information, in this thesis the target's 3D dynamic information in the world coordinate can be estimated to give better results for autonomous vehicle navigation.

From the work of [23], Chang et. al. propose the idea of velocity compensation. In their work, they integrate the image velocity optical flow vectors (local motion) and the camera motions (global motion) together for the motion compensation. This velocity compensation idea is very helpful to estimate the target's 3D world coordinate dynamics from a moving camera platform, which will be discussed later in this thesis.

Thirdly, as the autonomous vehicle is working in both the structured and unstructured environment, it is not always applicable to put additional sensors in the environment to help autonomous vehicles track the objects and navigate the environment. Therefore to make more general applications in different environments, the tracking system can only rely on the vehicle's onboard sensors to detect and track the object's 3D dynamics.

Generally, the object's motion properties are independent of the object's other visual features or models. Since the target tracking system in this thesis is proposed to track

the object in an unknown environment without any priori knowledge, the motion based algorithms can be used to detect and track the moving target. According to the literature reviews, the prerequisites for effectively tracking moving targets using CCD cameras generally include the following stages: detection of motion (motion estimation, motion segmentation and classification) and target tracking. Therefore the tracking scheme proposed in this thesis should also follow these procedures. First, the tracking system needs to detect the motion vectors from the images. Secondly, the classification methods are used to segment and cluster the images to find the major motion regions. Thirdly, the target's visual features are localized from the classified images through the whole image sequence. Finally, target tracking algorithms are used to estimate the target's dynamic states for every time stamp.

Concluded from literature reviews of visual target tracking technologies, optical flow based tracking algorithms have a number of advantages over the other tracking techniques. These include tracking with the moving cameras and minimal prior knowledge of the environment and target (no special features and no special tracking models) [89]. Optical flow methods can distinguish different objects by their different motion properties [88]. Since both the objects and the cameras are moving, optical flow vectors can be used for motion compensation to estimate the target's $3D$ world coordinate velocity from the relatively $2D$ motion fields [90]. By integrating these advantages with other techniques such as the motion classification, template matching or the object's motion models, it is possible to implement a robust tracking system working under different conditions. Also by fusing various visual cues and various sensor data a more reliable object tracking system could be designed for autonomous vehicle navigation and due to the increase in the object information the extension could be made to track $3D$ objects with stereo images [90].

One major limitation of the optical flow based tracking systems is the high computational cost of the algorithms and specialized hardware may be necessary for the complex real-time solutions. However nowadays the computer hardware technologies have a tremendous development, the computational time issue will effect the tracking system design less and less. So in this thesis the optical flow vectors can just be used as the main source for identifying and tracking the moving objects.

From the literature reviews each target tracking method has its own advantages and disadvantages. It is necessary to integrate the merits and ideas of each method into one tracking scheme to make a robust target tracking. In this thesis the target tracking scheme is proposed mainly based on the image motion vectors (optical flow vectors). With the motion vectors the tracking scheme can be developed based on the ideas and methodologies from different target tracking methods.

Data fusion will improve the system performance significantly and make robust estimations for both the motion detection and target tracking. Normally data fusion algorithms can be developed in two different ways: (1) fusing data from different sensors or different data from one sensor; (2) integrating different methods. So in the proposed target tracking scheme the ideas of data fusion can be introduced into the following stages.

- Fusion methodologies from the classification and inference methods can be introduced into the development of the motion segmentation and localization algorithms.
- Fusion methodologies from the estimation methods can be introduced into the development of the target's 3D dynamics estimation and state tracking algorithms.

With the successful data fusion the tracking system can easily and accurately achieve the tracking requirements for autonomous vehicle navigation.

2.5.1 Remaining Research Challenges

The following specific research challenges are identified for this thesis.

- In order to deal with both the structured and unstructured environments, a navigation algorithm based on visual target tracking should be developed as a better solution for autonomous vehicle navigation. Some of the applications of such a system are the autonomous vehicle target following, obstacle avoidance, path planning and so on.

- The tracking system should work in both the structured and unstructured situations. There are usually no additional sensors equipped in these environments. The proposed target tracking scheme can only rely on the vehicle's onboard sensors.
- Due to large amount of the visual data, it is very difficult to realize the real-time processing on the hardware placed in small limited hulls in autonomous vehicles. Fast and robust tracking algorithms are required for the real-time performance.
- In most visual tracking algorithms the camera system is stationary. Here a dynamically changing background is considered. Both the object and autonomous vehicle are moving. The target tracking system proposed in this thesis should be developed in the MCMO state.
- Instead of using the target's dynamics relative to the cameras, a tracking algorithm which can accurately derive the target's *3D* dynamic information in the world coordinate should be developed to give more applicable and accurate target tracking results.
- At present, many tracking algorithms are achieved by placing a specific visual property on the objects of interest. The limitation is mainly due to the finite object features or the models information. The research in this thesis aims at tracking the targets without placing any special property on the objects of interest for the autonomous vehicle navigation tasks.
- Since optical flow vectors are chosen as the main source to identify and track the object in the proposed target tracking scheme, it is important to solve two following problems.
 1. Optical flow vectors reflect the image motion properties. Different objects will have different motion regions and also there are background motions. Then the image processing algorithms which can accurately identify the region of interest from the optical flow vectors field should be developed.
 2. The system needs to gain an understanding of the target's real world *3D* dynamics from the target's estimated *2D* visual features. The *2D* optical flow vectors field can give the velocity and position information of the object

relative to the moving cameras. At the same time the autonomous vehicle's inertial sensors detect the vehicle's velocity and position in the world coordinate. So it is necessary to develop a fusion method to fuse the visual sensor's data with the vehicle's motion sensor data for estimating the target's 3D position and velocity in the world coordinate.

- The tracking system works in a Moving Camera and Moving Object (MCMO) state. The visual features of the object and the background are changing continuously. It is necessary to make the system have the ability to adapt to these changes, which means that the tracking algorithms should have some learning or adaptive estimation strategies.
- Finally in order to accurately track the object's 3D dynamics in different situations, there are several well-known tracking problems to be solved.
 1. First, normal objects do not have uniform or simple linear dynamic properties. The tracking will be difficult when the objects are with highly nonlinear or unpredictable motion properties. More complex and adaptive dynamic models should be developed to approximate the target's motion properties.
 2. Secondly, the proposed system is based on sensor data fusion. The vehicle's inertial motion sensors and vision sensors have different data capturing rates. Also since the target's 3D visual features need to be obtained from different image processing algorithms, the computational time for each method is different and usually quite large. Therefore for the fusion algorithm all the sensor data can not arrive at the same time. There is the Out-Of-Sequence Measurements "OOSM" (or called delayed measurements) problem. In order to make the accurate estimation, the OOSM problem must be considered.
 3. The occluded object tracking is one of the challenging problems in visual target tracking. In this thesis solutions need to be provided to solve the occlusion problems for the autonomous vehicle target tracking.

In this thesis all the above research problems have been carefully studied. Solutions are also proposed to solve these problems to some extent, which make the proposed system perform well for the autonomous vehicle target tracking.

Chapter 3

Proposed Tracking System

3.1 Introduction

Based on the previous literature reviews, a novel visual target tracking scheme is developed in this thesis for autonomous vehicle navigation. The proposed tracking scheme solves the remaining research problems and meets the autonomous vehicle navigation requirements.

First compared with other methods the tracking system is designed with following two advantages:

1. Only the vehicle's onboard sensors are used to measure the target features.
2. The tracking system does not put any burden on the moving target.

Now with the use of the vehicle's onboard sensors, there are several major issues to be considered for the autonomous vehicle visual target tracking [43]. (These problems have been summarized in the previous Chapter. They are listed here to reemphasize the proposed tracking strategies.)

- First, as the vision sensors (CCD stereo cameras pair) are moving with the vehicle, the background changes continuously. With a changing background it is hard to identify the object's dynamics based only on the vision sensors.

- Secondly, for autonomous vehicles in the outdoor environment, the optical properties of the target may be unpredictable or inconsistent (such as the changes in the environmental lighting condition). As within the image, the target is to be identified using specific visual properties, changes in the environment pose great difficulties.
- Thirdly, previous visual target tracking systems estimate the relative position and velocity of the object to the vehicle. In order to make more general applications with the target's dynamic information and obtain more accurate estimation results, the target's 3D velocity and position need to be estimated in the world coordinate system.
- Fourthly, natural moving targets often have unpredictable, complex or highly non-linear motion dynamics. The normally used linear dynamic models can not make an accurate target dynamics tracking.
- Lastly, different sensors have different data capturing and processing time. As a result, the fusion algorithm receives different measurements in different time stamps. In order to make the accurate target state estimation, this Out-Of-Sequence Measurements (OOSM) problem must be solved.

In the next section the novel target tracking methodology of the proposed scheme is explained and the step by step solutions are presented to solve the remaining problems.

3.2 Proposed Tracking System

The proposed tracking scheme has the ability to track the target's 3D dynamics in an unknown environment in the MCMO state. From the literature reviews, the prerequisites for effectively tracking the moving target using CCD cameras generally include detection of features (feature segmentation and classification) and target tracking. In such a way the tracking system is designed based on the follow strategies.

- First, in order to accurately identify the target from the image sequences with a changing background, the image motion properties (optical flow vectors) are

extracted from the image sequence. The system is designed to track the target's 3D dynamics, so the target's 3D visual depth is estimated from the stereo images.

- Secondly, different objects have different visual features, for example the motion properties. There are also background features together with the target's features. In the proposed visual target tracking scheme, these visual features are classified and localized by the image processing algorithms such as image clustering and template matching to find the useful ones.
- Thirdly, in order to track the target's 3D dynamics in the world coordinate, the target 3D visual features are fused with the vehicle's motion data. With the target's velocity and position estimates, the tracking filters such as Extended Kalman Filter are used for target tracking.

In detail, this thesis proposes a data fusion based target tracking scheme to solve the remaining problems mentioned in the previous section. Data fusion can be normally achieved either by using multiple (redundant) sensors or by combining several measurements of a single sensor [173]. In this thesis different data fusion strategies are formulated as follows for the proposed target tracking scheme.

- Initially, different visual features are fused to accurately extract the region of interest (ROI) from the image sequence with a changing background.
- Then, the idea of optical flow based motion compensation [23] is applied in the proposed target tracking scheme. Here the kinematic model of the autonomous vehicle is used to estimate the velocity of the cameras based on the vehicle's inertial sensors. Then optical flow vectors are fused with the moving camera's motion parameters based on the motion compensation approach. In such a way the target's 3D velocity and position are estimated in the world coordinate for the MCMO target tracking.
- Next, several simple and basic linear dynamic models are fused together through IMM to make the approximation of the target complex and nonlinear dynamic motion properties [90]. The fusion of different linear dynamic models through IMM makes the target dynamics estimation more robust.

Finally, for the OOSM problem the 1 – step – lag solution from [6] is integrated with the Extended Kalman Filter for the proposed tracking system to make better estimations.

Figure 3.1: The Proposed Tracking System

The proposed visual target tracking scheme is described schematically in Fig.3.1. The following chapters are organized in the order of this system flow chart. Each chapter (from Chapter 4 to Chapter 8) describes one component of the system and solves some of the remaining research problems. The experimental results are also shown in each Chapter to demonstrate the performance of the proposed approaches.

In the second step of the process (Chapter 4), the image optical flow vectors and stereo disparity are extracted from the image sequence. Then in the third step (Chapter

5), these visual features are fused for the motion field segmentation and clustering. The segmentation can be done with the simple K-means or the proposed unsupervised Bayesian on-line learning based segmentation algorithm. From the experimental testing the unsupervised Bayesian on-line learning based method gives better performance than the other methods.

In the fourth step (Chapter 6), the template matching and updating algorithm is developed to localize the region of interest from the clustered optical flow vectors fields. The template matching is between the target's optical flow vectors region and the whole image flow vectors field. Here the matching template is updated frame by frame. The template matching can also be done with color images using the methods such as the Bayesian fusion based template matching proposed in this thesis. With color images the matching is more accurate than the matching using only the motion vectors.

In the fifth step (Chapter 7), optical flow vectors are fused with the target's depth disparity information and then combined with the kinematic model of the camera system to estimate the target's 3D position and velocity in the world coordinate. These information is used later for the target dynamics tracking.

In the final step (Chapter 7), the target 3D dynamics is estimated with Extended Kalman Filter. Through the Extended Kalman Filter measurement equation the sensor data fusion is achieved for the proposed visual target tracking scheme. Here several simple and basic linear dynamic models with the IMM approach are used to make the approximation of the target complex motion properties. With the target tracking dynamic model, the *1-step-lag* OOSM estimation solution is integrated with the Extended Kalman Filter to solve the delayed measurement (OOSM) problem. The IMM approach and OOSM solution make the target dynamics tracking more robust.

Chapter 8 presents the experimental results obtained under different situations with nature image sequences. Chapter 9 concludes the research work in this thesis and makes the suggestions for future work.

Chapter 4

Visual Feature Extraction

In this Chapter visual feature extractions from the images are presented as the first step for the whole visual target tracking scheme. There are many visual features available from the images, such as color, texture, motion and so on. It is necessary to extract the useful ones from the images. The proposed tracking scheme in this thesis is mainly based on the object's dynamic properties, so first optical flow vectors estimation algorithms are explained. Next, the stereo disparity algorithm is described to obtain the target's 3D visual depth. Other useful visual features such as the image color or pixels' spatial location information can be read directly from the images.

4.1 Optical Flow Estimation

Motion estimation, which may refer to the image-plane motion (2D motion) or the object-motion (3D motion) estimation, is one of the fundamental problems in the computer vision and image processing research areas. Optical flow is one of the main solutions for the motion estimation. Optical flow vectors are defined as the temporal rate of changes of the image-plane coordinate, $(v_1, v_2) = (dx_1/dt, dx_2/dt)$ at a particular point $(\vec{x}, t)$ in the image as determined by the spatio-temporal variations of the image intensity pattern.

Numerous methods for optical flow vectors estimation have been developed in the last 20

years. Barron et. al. [10] analyze and compare several techniques that had been proposed before 1994, including four differential-based methods, two region-based methods, one energy-based method and two phase-based methods. More recent techniques can be found in [51] [52]. Among them, the three-dimensional (3D) structure tensor technique has been shown to have the low systematic error and low noise sensitivity and to yield very good results.

Despite the differences in all these methods, many of these techniques can be viewed conceptually in terms of three stages of processing:

- Pre-filtering or smoothing with the low-pass/band-pass filters in order to extract the signal structure of interest and to enhance the signal-to-noise ratio.
- The extraction of basic measurements, such as the spatio-temporal derivatives (to measure normal components of velocity) or the local correlation surface.
- The integration of these measurements to produce a 2D flow field, which often involves assumptions about the smoothness of the underlying flow field [10].

In this thesis, several commonly used techniques are tested for the estimation of optical flow vectors.

4.1.1 Horn and Schunck's Method

The first instances of the optical flow vectors estimation use the first-order derivatives and are based on the image translation [10] [71], i.e.

$$I(\vec{x}, t) = I(\vec{x} - \vec{v}t, 0) \quad (4.1)$$

where $\vec{v} = (u, v)^T$ and $\vec{x}$ corresponds to the 2D pixel's coordinates. From a Taylor expansion of Eq.(4.1) or more generally from an assumption that the intensity is considered, $dI(\vec{x}, t)/dt = 0$, the gradient constraint equation is easily derived:

$$\nabla I(\vec{x}, t) \cdot \vec{v} + I_t(\vec{x}, t) = 0 \quad (4.2)$$

where $I_t(\vec{x}, t)$ denotes the partial time derivation of $I(\vec{x}, t)$, $\nabla I(\vec{x}, t) = (I_x(\vec{x}, t), I_y(\vec{x}, t))^T$ and $\nabla I \cdot \vec{v}$ denotes the usual dot product. In effect, Eq.(4.2) yields the normal component of motion of spatial contours of the constant intensity, $\vec{v}_n = s\vec{n}$. The normal speed s and normal direction $\vec{n}$ are given by

$$s(\vec{x}, t) = \frac{-I_t(\vec{x}, t)}{\|\nabla I(\vec{x}, t)\|} \quad \vec{n}(\vec{x}, t) = \frac{\nabla I(\vec{x}, t)}{\|\nabla I(\vec{x}, t)\|} \quad (4.3)$$

There are two unknown components of $\vec{v}$ in Eq.(4.2), constrained by only one linear equation. Further constraints are therefore necessary to solve for both the components of $\vec{v}$. Some developed methods like Horn and Schunck's method are based on how to make further constraints to calculate the optical flow vectors.

Horn and Schunck's gradient-based optical flow estimation method [10] combines the image gradient constraint as given in (4.4)

$$\nabla I(\vec{x}, t) \cdot \vec{v} + I_t(\vec{x}, t) = 0 \quad (4.4)$$

with a global smoothness term to constrain the estimate of the optical flow vectors field $\vec{v} = (u(\vec{x}, t), v(\vec{x}, t))$ minimizing

$$\int_D (\nabla I \cdot \vec{v} + I_t)^2 + \lambda^2 (\|\nabla u\|_2^2 + \|\nabla v\|_2^2) dx \quad (4.5)$$

which is defined over a domain D , the magnitude of λ reflects the influence of the smoothness term, $I_t(\vec{x}, t)$ denotes the partial time derivative of the image intensity matrix $I(\vec{x}, t)$, $\nabla I(\vec{x}, t) = (I_x(\vec{x}, t), I_y(\vec{x}, t))^T$ and $\nabla I \cdot \vec{v}$ denotes the usual dot product.

Then Horn and Schunck's optical flow iterative estimation equations are given by Eq.(4.6) and Eq.(4.7).

$$u^{k+1} = \bar{u}^k - \frac{I_x[I_x \bar{u}^k + I_y \bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \quad (4.6)$$

$$v^{k+1} = \bar{v}^k - \frac{I_y[I_x \bar{u}^k + I_y \bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \quad (4.7)$$

where k denotes the iteration number, I_x and I_y denote the x and y direction derivations

of the image intensity matrix, u^0 and v^0 denote initial velocity estimates which are set to zero and $\bar{u}^k$ and $\bar{v}^k$ denote neighborhood averages of u^k and v^k .

Here the image intensity derivation matrixes I_x , I_y and I_t can be calculated by approximating them through the respective forward or backward finite difference. Horn and Schunck's method [71] is averaging four finite differences to obtain I_x , I_y and I_t as follows:

$$\begin{aligned}\frac{\partial I(x, y, t)}{\partial x} &\doteq \frac{1}{4} \{ I(n_1 + 1, n_2, k) - I(n_1, n_2, k) + I(n_1 + 1, n_2 + 1, k) - I(n_1, n_2 + 1, k) + \\ &\quad I(n_1 + 1, n_2, k + 1) - I(n_1, n_2, k + 1) + I(n_1 + 1, n_2 + 1, k + 1) - I(n_1, n_2 + 1, k + 1) \} \\ \frac{\partial I(x, y, t)}{\partial y} &\doteq \frac{1}{4} \{ I(n_1, n_2 + 1, k) - I(n_1, n_2, k) + I(n_1 + 1, n_2 + 1, k) - I(n_1 + 1, n_2, k) + \\ &\quad I(n_1, n_2 + 1, k + 1) - I(n_1, n_2, k + 1) + I(n_1 + 1, n_2 + 1, k + 1) - I(n_1 + 1, n_2, k + 1) \} \\ \frac{\partial I(x, y, t)}{\partial t} &\doteq \frac{1}{4} \{ I(n_1, n_2, k + 1) - I(n_1, n_2, k) + I(n_1 + 1, n_2, k + 1) - I(n_1 + 1, n_2, k) + \\ &\quad I(n_1, n_2 + 1, k + 1) - I(n_1, n_2 + 1, k) + I(n_1 + 1, n_2 + 1, k + 1) - I(n_1 + 1, n_2 + 1, k) \} \end{aligned} \quad (4.8)$$

The original method described [71] by Horn and Schunck uses first-order differences to estimate intensity derivatives. Because this is a relatively crude form of numerical differentiation and can be the source of considerable errors, the methods with spatio-temporal pre-smoothing and 4-point central differences for differentiation (with mask coefficients) are also implemented here. A spatio-temporal Gaussian pre-filter is used with a standard deviation of 1.5 pixels in space and 1.5 frames in time (1.5 pixels-frames), sampled out to three standard deviations. And this is the so-called modified Horn and Schunck's method [222].

4.1.2 Lucas and Kanade's Method

Following Lucas and Kanade [128] and others [10] methods, a weighted least-squares (LS) fit of the local first-order constraints Eq.(4.2) is implemented to a constant model for the optical flow vectors field $\vec{v}$ in each small spatial neighborhood Ω by minimizing

$$\sum_{\vec{x} \in \Omega} W^2(\vec{x}) [\nabla I(\vec{x}, t) \cdot \vec{v} + I_t(\vec{x}, t)]^2 \quad (4.9)$$

where $W(\vec{x})$ denotes a window function that gives more influence to the constraints at the center of the neighborhood than those at the periphery. The solution to Eq.(4.9) is given by

$$A^T W^2 A \vec{v} = A^T W^2 b \quad (4.10)$$

where, for n points in $x \in \Omega$ at a single time t ,

$$A = [\nabla I(\vec{x}_1), \dots, \nabla I(\vec{x}_n)]^T, \quad W = \text{diag}[W(\vec{x}_1), \dots, W(\vec{x}_n)], \quad b = -(I_t(\vec{x}_1), \dots, I_t(\vec{x}_n))^T \quad (4.11)$$

The solution to Eq.(4.11) is

$$\vec{v} = [A^T W^2 A]^{-1} A^T W^2 b \quad (4.12)$$

which is solved in the closed form when $A^T W^2 A$ is nonsingular, since it is a 2×2 matrix:

$$A^T W^2 A = \begin{bmatrix} \sum W^2(\vec{x}) I_x^2(\vec{x}) & \sum W^2(\vec{x}) I_x(\vec{x}) I_y(\vec{x}) \\ \sum W^2(\vec{x}) I_y(\vec{x}) I_x(\vec{x}) & \sum W^2(\vec{x}) I_y^2(\vec{x}) \end{bmatrix} \quad (4.13)$$

where all sums are taken over the points in the neighborhood Ω .

Equations (4.9) and (4.10) may also be viewed as the weighted least-square estimates of $\vec{v}$ from the estimates of normal velocities $\vec{v}_n = s\vec{n}$; i.e. Eq.(4.9) is equivalent to

$$\sum_{\vec{x} \in \Omega} W^2(\vec{x}) \omega^2(\vec{x}) [\vec{v} \cdot \vec{n}(\vec{x}) - s(\vec{x})]^2 \quad (4.14)$$

where the coefficient $\omega^2(\vec{x})$ reflects the confidence in the normal velocity estimates; here $\omega(\vec{x}) = ||\nabla I(\vec{x}, t)||$.

4.1.3 The Three-Dimensional (3D) Structure Tensor Technique for Optical Flow Estimation

In the recent publication [125], the 3D structure tensor technique has been shown to have the low systematic error and low noise sensitivity for optical flow vectors estimation.

Extending the constant optical flow model to a parametric model can further improve the accuracy of optical flow estimates. In [125] the optical flow estimation problem is converted to the generalized eigenvalue analysis. Moreover, the confidence measures defined by the generalized eigenvalues can be used to adaptively adjust the coherent motion region and achieve a more accurate and robust result.

The main assumption underlying the 3D structure tensor technique is the brightness change constraint, which assumes that the image brightness changes only due to motion. To regularize the results, it constructs the tensor for each pixel within its neighborhood, in which the local optical flow is assumed to be constant. It then converts the optical flow estimation problem to an eigenvalue analysis problem. The following method is taken from the optical flow estimation algorithm developed in [125].

According to the commonly-used assumption that the image brightness $g(\vec{x}, t)$ at a point $\vec{x} = [x, y]^T$ and the time t change only due to motion, there is

$$\frac{dg}{dt} = \frac{dg}{dx} \frac{dx}{dt} + \frac{dg}{dy} \frac{dy}{dt} + \frac{dg}{dt} = 0 \quad (4.15)$$

Let the optical flow $\vec{f} = [f_x, f_y]^T = [dx/dt, dy/dt]^T$, $(\nabla_{\vec{x}t}g) = [g_x, g_y, g_t]^T = [dg/dx, dg/dy, dg/dt]^T$, Eq.(4.15) leads to the optical flow constraint equation

$$\begin{aligned} [g_x, g_y]^T \vec{f} + g_t &= 0 \Rightarrow [g_x, g_y, g_t][f_x, f_y, 1]^T = 0 \\ \Rightarrow v_t^{-1}[g_x, g_y, g_t][v_x, v_y, v_t]^T &= 0 \Rightarrow [g_x, g_y, g_t][v_x, v_y, v_t]^T = 0 \\ &\Rightarrow (\nabla_{\vec{x}t}g)^T \vec{v} = 0 \end{aligned} \quad (4.16)$$

where

$$f_x = \frac{v_x}{v_t}, \quad f_y = \frac{v_y}{v_t}, \quad \vec{v} = [v_x, v_y, v_t]^T \quad (4.17)$$

Here, $\vec{v}$ is the 3D flow in the $\vec{x}t$ - space. It can be solved by minimizing the following energy function in the least-squares sense:

$$E(\vec{x}) = \int_{-\infty}^{\infty} \omega(\vec{x} - \vec{x}') [(\nabla_{\vec{x}t}g)^T \vec{v}]^2 d\vec{x}' \quad (4.18)$$

where $\omega(\cdot)$ is a window function that defines the neighborhood N around $\vec{x}$.

Define the mean operation “ $\langle \cdot \rangle$ ” as follows

$$\langle f \rangle = \int_{-\infty}^{\infty} \omega(\vec{x} - \vec{x}') f d\vec{x}' \quad (4.19)$$

The energy function in Eq.(4.18) can be expressed as

$$E(\vec{x}) = \langle [(\nabla_{\vec{x}t} g)^T \vec{v}]^2 \rangle = \langle \vec{v}^T (\nabla_{\vec{x}t} g) (\nabla_{\vec{x}t} g)^T \vec{v} \rangle \quad (4.20)$$

Instead of assuming that the 3D flow $\vec{v}$ is constant within the neighborhood N , the affine model is used as the parametric model of the 3D flow $\vec{v}$:

$$\vec{v} = \begin{bmatrix} a'_1 & a'_2 & a'_3 \\ a'_4 & a'_5 & a'_6 \\ a'_7 & a'_8 & a'_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ t \end{bmatrix} + \begin{bmatrix} a'_{10} \\ a'_{11} \\ a'_{12} \end{bmatrix} \quad (4.21)$$

For a particular frame, t is a constant and the last two terms of each of the above equations can be combined into one term. The affine model can be simplified as follows:

$$\begin{aligned} \vec{v} &= \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_7 \\ a_8 \\ a_9 \end{bmatrix} \\ &= \begin{bmatrix} x & y & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & x & y & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & x & y & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_9 \end{bmatrix} = B\vec{p} \end{aligned} \quad (4.22)$$

Noting that within the neighborhood N the parameter $\vec{p}$ rather than $\vec{v}$ is constant, and substituting $\vec{v}$ in Eq.(4.22) into Eq.(4.20), it has

$$\begin{aligned} E(\vec{x}) &= \langle [(\nabla_{\vec{x}t} g)^T B\vec{p}]^2 \rangle = \langle \vec{p}^T (B^T (\nabla_{\vec{x}t} g) (\nabla_{\vec{x}t} g)^T B) \vec{p} \rangle \\ &= \vec{p}^T \langle (B^T (\nabla_{\vec{x}t} g) (\nabla_{\vec{x}t} g)^T B) \rangle \vec{p} = \vec{p}^T T \vec{p} \end{aligned} \quad (4.23)$$

where

$$T = \langle B^T (\nabla_{\vec{x}t} g) (\nabla_{\vec{x}t} g)^T B \rangle \quad (4.24)$$

is the parametric structure tensor (affine in this case).

To avoid the trivial solution $\vec{v} = 0$ for Eq.(4.20), an additional constraint is needed. Noting that $\vec{v} = B\vec{p}$,

$$\vec{v}^T \vec{v} = 1 \Rightarrow \vec{p}^T B^T B \vec{p} = \vec{p}^T D \vec{p} = 1 \quad \text{where} \quad D = B^T B \quad (4.25)$$

Other options include setting $a_9 = 1$ in Eq.(4.22). By using Lagrange multipliers, the combined energy function becomes

$$L(\vec{p}, \lambda) = \vec{p}^T T \vec{p} + \lambda(1 - \vec{p}^T D \vec{p}) \quad (4.26)$$

Solving Eq.(4.26) it shows

$$\frac{\partial L(\vec{p}, \lambda)}{\partial p_k} = 0 \Rightarrow T \vec{p} = \lambda D \vec{p} \quad (4.27)$$

subject to

$$\frac{\partial L(\vec{p}, \lambda)}{\partial \lambda} = 0 \Rightarrow \vec{p}^T D \vec{p} = 1 \quad (4.28)$$

where $\vec{p} = [p_k]$ and $k = 1, 2, \dots, 9$. Therefore, the minimization problem Eq.(4.18) is converted to the generalized eigenvalue problem as in Eq.(4.27).

Where the optical flow is constant within the neighborhood, the generalized eigenvalue problem reduces to an ordinary eigenvalue problem

$$T \vec{v} = \lambda \vec{v} \quad (4.29)$$

subject to

$$\vec{v}^T \vec{v} = 1 \quad (4.30)$$

where $T = \langle (\nabla_{\vec{x}t} g) (\nabla_{\vec{x}t} g)^T \rangle$.

The motivation for formulating the optical flow estimation problem as a generalized

eigenvalue problem rather than an ordinary linear square problem is that it essentially provides a total least-squares (or error in variables) solution, which assumes that both the spatial and temporal image solutions are noisy.

Notice that in Eq.(4.27) both the matrixes $T_{9 \times 9}$ and $D_{9 \times 9}$ are real, symmetric, and nonnegative-definite. According to the simultaneous diagonalization theorem, when D is nonsingular, there exists a matrix $P = [\vec{p}_1, \vec{p}_2, \dots, \vec{p}_9]$ that diagonalizes both

$$P^T T P = \text{diag}(t_1, t_2, \dots, t_9), \quad P^T D P = \text{diag}(d_1, d_2, \dots, d_9) \quad (4.31)$$

In addition, $T\vec{p}_i = \lambda_i D\vec{p}_i$ for $i = 1, 2, \dots, 9$ where $\lambda_i = t_i/d_i$. When D is singular, only n of the d_i s are nonzero, where n is the rank of D . In the problem of this method, $n = 3$, i.e., there are only three defined eigenvalue/eigenvector pairs.

Left-multiplying both sides of Eq.(4.29) by $\vec{p}_i^T$ and using Eq.(4.22), it has

$$\begin{aligned} \vec{p}_i^T T \vec{p}_i &= \lambda_i \vec{p}_i^T D \vec{p}_i \\ \Rightarrow \lambda_i &= \frac{\vec{p}_i^T T \vec{p}_i}{\vec{p}_i^T D \vec{p}_i} = \frac{\vec{p}_i^T \langle B^T (\nabla \vec{x}_{tg}) (\nabla \vec{x}_{tg})^T B \rangle \vec{p}_i}{\vec{p}_i^T B^T B \vec{p}_i} \\ &= \frac{\| \langle (\nabla \vec{x}_{tg})^T B \vec{p}_i \rangle \|^2}{\| B \vec{p}_i \|^2} = \frac{\| \langle (\nabla \vec{x}_{tg})^T \vec{v}_i \rangle \|^2}{\| \vec{v}_i \|^2} \geq 0 \end{aligned} \quad (4.32)$$

Here λ_i is defined only when $\vec{v}_i = B\vec{p}_i \neq 0$, which is guaranteed by the constraint Eq.(4.28).

Without the loss of generality, let's first order the eigenvalues as

$$0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \quad (4.33)$$

with their corresponding eigenvectors denoted by $\vec{p}_1$, $\vec{p}_2$ and $\vec{p}_3$.

Substituting Eq.(4.27) into Eq.(4.23), and with the constraint that $\vec{v}^T \vec{v} = 1$, it shows

$$\begin{aligned} E(\vec{x}) &= \vec{p}^T T \vec{p} = \lambda \vec{p}^T D \vec{p} = \lambda \vec{v}^T \vec{v} = \lambda \\ \Rightarrow \min(E(\vec{x})) &= \min(\lambda_k)_{k=1,2,3} = \lambda_1 \end{aligned} \quad (4.34)$$

i.e., the energy function E is minimized when the affine parameter $\vec{p}$ is the generalized eigenvector $\vec{p}_1$ of the affine structure tensor T that corresponds to the minimum generalized eigenvalue λ_1 . Once $\vec{p}_1$ is obtained, the optical flow can be calculated from Eq.(4.17) and Eq.(4.22) by substituting $\vec{p}_1$ for $\vec{p}$.

The method from [125] is an adaptive optical flow vectors estimation approach. In their method the confidence measure derived from the generalized eigenvalues is used to adaptively adjust the coherent motion region to further improve the accuracy.

4.2 Visual Depth from Stereo Imaging

4.2.1 Introduction

Extraction of the three-dimensional ($3D$) structures of a scene from stereo images has been studied by the computer vision community for decades [18].

Stereo refers to the problem of determining the $3D$ structure of a scene from two or more images taken from distinct viewpoints [54]. The fundamental basis for stereo is that a single $3D$ physical location projects to a unique pair of image locations in two observing cameras. As a result if the two cameras images are available and the image locations that correspond to the same physical point in space can be determined, then it is possible to estimate its $3D$ location.

The primary problems to be solved in stereo are the calibration, correspondence, and reconstruction [18]. Here the correspondence problems are normally the main research topics in the research community. In practice given two images and from the information contained in these images, the disparities must be calculated. The correspondence problem consists of determining the locations in each camera image that are the projections of the same physical point in space. No general solution to the correspondence problem exists, due to ambiguous matches (e.g., due to occlusions, secularities, or lack of texture). Thus, a variety of constraints (e.g., epipolar geometry) and assumptions (e.g., image brightness constancy and surface smoothness) are commonly exploited to make the problem tractable.

As noted previously, stereo disparities may be estimated in a number of ways and by exploiting a number of constraints. All of these methods attempt to match pixels in one image with their corresponding pixels in the other image. For simplicity, the constraints on a small number of pixels surrounding a pixel of interest are referred as local constraints. Similarly, the constraints on scan-lines or on the entire image are loosely referred as global constraints. Local methods can be very efficient, but they are sensitive to locally ambiguous regions in images (e.g., occlusion regions or regions with uniform textures.). Global methods can be less sensitive to these problems since global constraints provide additional supports for regions difficult to match locally, but these global methods are more computationally expensive.

The local methods fall into three broad categories: block matching, gradient methods and feature matching. The global correspondence methods exploit nonlocal constraints in order to reduce the sensitivity to local regions in the image that fail to match, due to the occlusion, uniform texture, etc. The use of these constraints makes the computational complexity of global matching significantly greater than that of local matching. By far, the most common approach to global matching is the dynamic programming, which uses the ordering and smoothness constraints to optimize the correspondences in each scan line. Two variants of dynamic programming are the standard method and the one based on Tomasi and Manduchi's intrinsic curves [18]. More recent graph cut, diffusion, and belief propagation methods, all of which incorporate more general two-dimensional local cohesion constraints, can also be used.

After roughly 30 years of research on computational stereo (in the computer vision community), many elements of stereo algorithms are well understood, especially for the accurate stereo calibration and efficient algorithms for the local correspondence. As a result during the past decade, the focuses have been turned from the fundamentals of stereopsis to more difficult problems, such as the global correspondence and methods for handling occlusions [18]. Perhaps the most practically significant advance in the last decade has been the appearance of real-time stereo systems, for example on special-purpose computers. Due to the large part to these real-time implementations, researches on real-time stereo applications have blossomed in the latter part of this decade. Real-time algorithms, however, are still relatively simplistic, and most of the global matching

and occlusion handling methods discussed do not currently run in real-time. More demanding potential applications (e.g., virtual reality) require both the real-time algorithms and very precise, reliable, and dense depth estimates. Solving the problems of stereo timing, precision and reliability jointly remains a challenging research topic that the rapid process can be predicted and seen in the coming decade.

4.2.2 Depth from Stereo

In this thesis, the image stereo disparity is estimated to obtain the target's 3D visual depth information. Here the basic stereo disparity theory and the commonly used SAD disparity estimation method are described.

The stereo images sequence is recorded using two CCD cameras. As shown in Fig.4.1

Figure 4.1: The Simple Camera Geometry for Stereo Images from Horn's book [71]

the two cameras are rigidly attached to each other so that their optical axes are parallel and separated by a distance b_1 . Here f is the camera's focal length. The difference of the same point in the left and right image coordinates $x_l' - x_r'$ is known as disparity which is denoted by d . Thus a point with (X, Y, Z) in the world coordinate can be obtained

from Eq.(4.35) as shown in Fig.4.1.

$$X = b_1 \frac{(x'_l + x'_r)/2}{x'_l - x'_r}; Y = b_1 \frac{(y'_l + y'_r)/2}{x'_l - x'_r}; Z = b_1 \frac{f}{x'_l - x'_r} \quad (4.35)$$

4.2.3 Disparity Estimation

For every image pixel in the left image the disparity estimation is to find the corresponding pixel in the right image (a match). Matching each single pixel between two images requires much more computational time. Therefore a small region containing the image pixels, called the correlation window, is used to make the correspondence matching faster. The window in this thesis is chosen to be the region of interest from the image segmentation and template matching results. This kind of choice may also have some drawbacks. A discussion of how to overcome them is beyond the scope of this thesis, seeing for example [18].

Without the loss of generality this research assumes the already calibrated cameras system in the sequel, i.e. the epipolar lines are parallel to the image rows. To obtain a first estimate of the disparity, an area based matching algorithm is used. For that purpose image blocks of the left and right stereo images are compared along the epipolar lines.

In this thesis the Sum of the Absolute RGB Difference (SAD) algorithm given in Eq.(4.36) is used to estimate the disparity d . The Sum of Absolute RGB Difference (SAD) performs better than the normally used Sum of Squared Difference (SSD) or the Zero-mean Sum of Squared Difference (ZSSD) in the presence of outliers. According to [146], using the color information instead of gray values improves the signal to noise ratio by 0.2 – 0.5.

$$SAD(x, y, d) = \sum_{i=-1/2(win_x-1)}^{1/2(win_x-1)} \sum_{j=-1/2(win_y-1)}^{1/2(win_y-1)} \left(\begin{aligned} &|R_l(x+i, y+j) - R_r(x+i+d, y+j)| \\ &+ |G_l(x+i, y+j) - G_r(x+i+d, y+j)| \\ &+ |B_l(x+i, y+j) - B_r(x+i+d, y+j)| \end{aligned} \right) \quad (4.36)$$

here R , G and B are the color components of the respective pixels, index l stands for the left image, and index r for the right one. The dimensions of the correlation window win_x and win_y have to be uneven numbers, otherwise there would be an offset of half a pixel between the disparity and the left image. In this thesis the correspondence matching is done only by matching the region of interest from one image onto the other to estimate the disparity value, which can reduce the computation time by ignoring the whole image matching.

4.2.4 Some Stereo Images

Here are the stereo images captured by the CCD cameras in both the indoor and outdoor environments.

4.2.1:

4.2.2:

4.2.3:

4.2.4:

Figure 4.2: Some Stereo Images in Both the Indoor and Outdoor Structured and Unstructured Environments. These images are used later to test the proposed target tracking system in Chapter 8.

The disparity calculation is obtained from these stereo images. The computation of the stereo disparities is done with Matlab. Some other existing softwares (commercial or free) also provide the stereo disparity estimation functions and can be used for the testing. Since in the proposed tracking scheme the disparity calculation is based only on the extracted region of interest, the whole disparity estimation result is not shown here. The estimated disparity is directly used to obtain the target's depth information, which is shown in Fig.(4.3).

The estimated stereo disparity is used later for the target's 3D dynamics estimation in Section 7. The estimated distance will be compared with the ground truth measured distance in the experimental section of this thesis to demonstrate the stereo disparity

Figure 4.3: The distance estimation result. This figure shows the estimated visual depth vs. the frame number for the stereo images of Fig.(4.2.1) and Fig.(4.2.2). The x axis is the image frame number. The y axis is the estimated distance (cm) between the cameras and the object.

estimation performance.

4.3 Experimental Results of Optical Flow Estimation

Now after all the three commonly used optical flow algorithms are described as above, the experimental comparisons are done in this part.

The first step in calculating the optical flow vectors is to compute the gradient of the image sequence in the $3D \vec{x}t$ - space. However, the commonly-used first-order discrete difference operators, such as the forward/backward/centered differences, are only very crude approximations to an ideal first-order derivative operator. Such operators introduce large errors in the gradient, and thus propagate large deviations to the optical flow estimates. Better derivative operators include series-designed and B-spline based deriva-

tive operators. Both of them provide larger support regions for the accurate derivative computation. In the experiments here, a filter of radius 5 is used.

$$D = \frac{1}{250}[-225, -150, 600, -2100, 2100, -600, 150, 225] \quad (4.37)$$

This helps improve the accuracy of optical flow vectors estimation.

It is well known that appropriately smoothing the image frames improves the optical flow estimation [10]. In the experiments, a Gaussian filter with the standard deviation of 1.5 on the spatial axis and 1.5 on the temporal axis is used to smooth the image sequences to alleviate the aliasing problem. With the Gaussian filtering, first, the optical flow estimation accuracy will be improved. Secondly, this smoothing can reduce the textured object effects and make the following image segmentation and clustering more accurate.

Normally the gray level images are used for optical flow vectors estimation. Because when the color information of the images is discarded, the time for optical flow vectors calculation can be greatly reduced.

4.4.1:

4.4.2:

4.4.3:

4.4.4:

Figure 4.4: One person moving in the outdoor unstructured environment. The first three figures are the color images. Fig.(4.4.4) is the gray level image for the optical flow estimation.

Fig.(4.5) shows the optical flow vectors field estimated from Lucas and Kanade's method.

From Fig.(4.6) the optical flow vectors field from Horn and Schunck's method is very dense. The 100% dense optical flow vectors field is good for the following image clustering and segmentation calculation. The movement of the person in Fig.(4.6) is also much clearer than the one in Fig.(4.5). It is one of the main reasons why Horn and Schunck's method is considered better than Lucas and Kanade's method for the proposed tracking

4.5.1:

4.5.2:

Figure 4.5: Optical flow vectors fields estimated from Lucas and Kanade's method

4.6.1:

4.6.2:

Figure 4.6: Optical flow vectors fields estimated from Horn and Schunck's method.

system.

As shown in Fig.(4.7) the platform carrying the camera is moving towards the moving target (the blue color target is pointed in Fig.(4.7.1)), and as a result the size of the

Figure 4.7: Indoor moving target with moving background. A blue color object is moving in the image sequence. It can be seen that the background is changing frame by frame.

object appearing in the scene increases from left to right. This image sequence is also used to test different optical flow estimation methods.

Fig.(4.8) shows the optical flow vectors fields estimated from Horn and Schunck's method. The optical flow field is 100% dense, which reflects the movements of the target and its background. Each vector reflects the image pixel's velocity. From these figures the optical flow vectors from Horn and Schunck's method are also very accurate to reflect the major motions in the image.

Some other comparisons of the three different optical flow estimation methods are presented next.

Fig.(4.10.1) shows the optical flow vectors field from the structure tensor method. Fig.(4.10.1) enlarges the optical flow vectors from Fig.(4.10.1). From this figure the flow vectors all point to one direction, which is the target moving direction in the image.

Fig.(4.11.2) enlarges the optical flow vectors field from Horn and Schunck's method (Fig.(4.11.1)). From this figure the optical flow vectors point to different directions. The image optical flow vectors cannot accurately reflect each image pixel's correct moving direction and velocity in the images.

Next the above experimental results are discussed. Based on the experimental comparisons and theoretical discussions, Horn and Schunck's method is selected for the proposed target tracking scheme in this thesis.

4.8.1:

4.8.2:

4.8.3:

4.8.4:

Figure 4.8: Optical flow vectors fields from the image sequence of Fig.(4.7) with the modified Horn and Schunck's method. Figures (4.8.3) and (4.8.4) are two of the enlarged ones for clarity.

4.4 Summary

In this section experimental results on optical flow estimation are discussed. The reasons of selecting Horn and Schunck's method for the proposed visual target tracking system are explained.

It is clear from Horn and Schunck's method as shown in Fig.(4.11) that the optical flow

Figure 4.9: The black area is moving in the images to test the optical flow estimation algorithms.

Figure 4.10: Optical flow vectors estimated from the structure tensor method for the image sequence of Fig.(4.9).

vectors in Fig.(4.10) are accurately estimated showing the moving target in the image sequence. This is because the structure tensor method solves the aperture problem better than Horn and Schunck's method [125]. However as seen in Fig.(4.10) there are more noises in the optical flow vectors field. And the biggest problem with the structure tensor method is that it computes the optical flow vectors very slowly. The structure tensor method calculates the eigen-value of each pixel from the image, which requires many iterations. For the optical flow estimation of one frame from [125], it takes about four minutes for the calculation. Even after some optimization and simplification of the optical flow vectors estimation program, the speed can only be reduced to 2 –

4.11.1:

4.11.2:

Figure 4.11: Optical flow vectors estimated from Horn and Schunck's Method for the Image Sequence of Fig.(4.9).

3mins/frame, which is still much slower than that by Horn and Schunck's method. So it is really difficult to do the fast or real-time processing using the structure tensor method.

The estimation results from Horn and Schunck's method is not as accurate as those from the structure tensor method to reflect the motion of the target pixels due to its sensitivity to the aperture problem. There are few noises in the optical flow vectors fields as shown in Fig.(4.11). There are also apparent optical flow vectors in the moving regions from Horn and Schunck's method and also much smaller or no motion vectors in the stationary parts. As in the proposed visual target tracking scheme, the optical flow vectors are only used to find the major motion regions in the images. The flow vectors field from Horn and Schunck's method is good enough for the proposed visual target tracking scheme to conduct the following image segmentation and clustering step. Also since the following image processing steps are developed with the visual features' data fusion, the little inaccuracy from Horn and Schunck's method slightly affects the features' extraction and identification.

Compared with other gradient-based optical flow estimation methods, Horn and Schunck's algorithm is more suitable for the proposed visual target tracking scheme due to the fol-

lowing reasons [10] [71]. First, Horn and Schunck's method is a global estimation method which aids the regions having insufficient local constraints benefit from the estimates at surrounding regions, while Lucas and Kanade's method is a local one. Secondly, Horn and Schunck's method uses the iterative Gauss-Seidel equations to prevent the propagation of roundoff errors and allow the minimization to be solved (whereas the Gaussian elimination would fail badly). Thirdly, Horn and Schunck's method is an iterative process which refines the optical flow estimation result. Finally and most importantly, Horn and Schunck's method is faster in computation from the experimental testing in this thesis.

If the optical flow estimation accuracy accepted by the tracking system, its faster computational speed and 100% estimated dense optical flow vectors field are considered, Horn and Schunck's method is chosen as the better one than the other optical flow estimation algorithms for the proposed visual target tracking scheme in this thesis.

However one issue is necessary to be considered for the implementation of Horn and Schunck's estimation method. Normally 20 iterations for Horn and Schunck's method is not enough to give good results, but at least 100 would be better. With 100 iterations it is better yet to check if the convergence has been obtained by seeing if the norms of the last and second last flow field difference are less than some threshold.

Chapter 5

Visual Features Classification: Image Segmentation and Clustering

After the visual features are extracted from the image sequence, it is necessary to classify these features and find the useful ones from the images, which is the third step of the proposed visual target tracking scheme. In this Chapter, the image segmentation and clustering methods are presented and the novel unsupervised Bayesian on-line learning based image clustering method is presented as a better solution for the proposed tracking scheme.

5.1 Introduction

The moving object's segmentation in the video images is a key process in a large number of multimedia and computer vision applications. It is also a major step in the proposed visual target tracking scheme. Here the image segmentation aims at detecting the regions corresponding to the moving objects. Detecting moving regions provides a focus of attention for the later processes such as the tracking and estimation because only these regions need to be considered in the later processes. Also different moving regions may correspond to different moving targets in the natural scenes. For instance, the image sequences captured by the moving cameras mounted on the autonomous vehicle probably

include humans, vehicles and other moving objects. To further track objects and analyze their behaviors, it is essential to correctly classify moving objects.

However, in the MCMO state presented in the proposed visual target tracking scheme, most of the difficulties associated with the image segmentation and clustering are due to the presence of a relative motion of the observer with respect to the motion of the objects and background. The segmentation of moving objects becomes difficult when the video is acquired by a moving camera with an unconstrained and a priori unknown motion. In this case, the segmentation by computing visual motion alone is not always reliable, and other features must be exploited such as the color, shape, texture and so on.

Several researchers have approached these problems of the object based image segmentation acquired by the moving cameras. The proposed solutions can be mainly differentiated into three classes:

1. based on the egomotion computation: these methods initially compute the egomotion and, after the compensation of the camera motion, they apply an algorithm defined for the fixed camera. [33];
2. based on the motion segmentation: the objects are mainly segmented by using the motion vectors computed at the image pixel level. [24];
3. based on the region merging with motion: the objects are obtained with a segmentation using several visual features, and then merged on the motion parameters computed on a region-level. [32] [193]

As discussed earlier the image segmentation algorithm in this thesis is considered as a classification approach, which is mainly in the second and third groups of the above methods. Given a set of image pixels the classification is to find the best correspondence of the image pixels with some property of the image visual features clusters (color, motion vectors, spatial locations, etc.). The segmentation is considered good if the clusters computed are homogeneous with respect to the feature vector, with the additional constraint that no two clusters should be similar (otherwise their pixels might be grouped together into the same cluster).

The most popular visual features used for the image clustering and segmentation are the motion vectors, color. At the same time texture [25] and geometric properties [116] are also widely used. Some of the techniques considered so far [31] [215] try to solve the image segmentation problem using only the motion information. This, however, ignores the wealth of the additional information that is present in the image intensity structure. The image structure and pixel motion can both be considered at the same time by assigning a combined score to each pixel and then finding the optimal segmentation based on all these properties. Several researchers have proposed the use of multiple features for the image segmentation [2] [13] [86] [89] [95] [193] [226]. These multiple visual features fusion algorithms solve the only motion or color based segmentation problems ([24] [31] [215]) and lead to robust image pixels clustering and segmentation. Thus this thesis proposes to use the pixel's color values, motion flow vectors and other visual features as the segmentation components and then combines their strengths by adjusting their weights for the accurate image pixels clustering.

In this Chapter two different image segmentation methods are presented and compared. The first one is the simple K-means image segmentation and the second one is the unsupervised Bayesian on-line learning based image pixels clustering.

One thing to be mentioned here is that a number of color spaces exist such as the RGB, YUV, HSV and so on, in which the segmentation of images can be performed. In this thesis the methods described next and the results reported later are all based on the use of the RGB (Red, Green, Blue) color space. The reason using the RGB color space is because the RGB colors can be easily and directly read from the images and each image pixel's RGB color values can be easily used for programming. However the image segmentation methods proposed in this Chapter could also be easily extended to any other color space.

Another thing to be mentioned is that some modern computer vision techniques use the image descriptors for image processing. If the algorithm proposed in this Chapter is also considered in the categories of the image descriptors techniques, then it probably belongs to the global image descriptors. The whole image's color, motion and spatial information are considered as a whole for processing. It is different from the local

descriptors techniques. The local descriptors are based on the selected interested points from the images for processing, which can only reflect some portions of the images.

5.2 Image Clustering and Segmentation Using the K-means Algorithm

If the segmentation of an image into K classes (clusters) is considered, the segmentation label (l) field assumes each image pixel vector $\vec{x}$ with one of the K values, i.e., $\vec{x}_l$, $l = 1, \dots, K$. In the case of the scalar features, such as the pixel intensities, the image clustering can be considered as a method of determining the $K - 1$ thresholds that define the decision boundaries in the $1D$ feature space. With the M -dimensional vector features, the segmentation corresponds to partitioning the M -dimensional features space into K regions. A standard procedure for clustering is to assign each sample to the class of the nearest cluster mean. In the unsupervised mode, this can be achieved by an iterative procedure, known as the K-means algorithm, because the cluster means are also initially unknown.

In the following, the K-means algorithm is described that an image is segmented into K clusters based on the fusion of the optical flow vectors, spatial location and color intensity values of the pixels.

Let $\vec{x}$ denote the feature vector of an image pixel. A translational motion model modelling the object motion in the image sequence is used, and thus it is able to use the optical flow estimates v_x and v_y directly as the motion parameters. The K-means clustering algorithm is used with the feature vector:

$$\vec{x} = [v_x, v_y, x, y, R, G, B] \quad (5.1)$$

where x, y are the pixel's locations to add the spatial information to the vector, and R, G, B are the intensities of the color components at that point. Here the moving target's optical flow vectors are fused with the position and color information into one feature vector. Fusing all these different components, the segmentation and clustering will be

more precise and can also reflect the different components clusters.

The K-means method aims to minimize the performance index

$$J = \sum_{l=1}^K \sum_{\vec{x} \in c_l^{(i)}} \|\vec{x} - \mu_l^{i+1}\|^2 \quad (5.2)$$

where $c_l^{(i)}$ denotes the set of samples assigned to the cluster l after the i th iteration, and μ_l denotes the mean of the l th cluster. This index measures the sum of the distances of each sample from their respective cluster means. The K-means algorithm, given by the following steps, usually converges to a local minimum of the performance index J .

5.2.1 K-means Algorithm

1. Choose K initial cluster means, $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_K^{(1)}$ arbitrarily.
2. At the i th iteration assign each pixel's vector, $\vec{x}$, to one of the K clusters according to the relation

$$\vec{x} \in c_j^{(i)} \quad \text{if } \|\vec{x} - \mu_j^{(i)}\| < \|\vec{x} - \mu_l^{(i)}\| \quad (5.3)$$

for all $l = 1, 2, \dots, K, l \neq j$, where $c_l^{(i)}$ denotes the set of samples whose cluster center is $\mu_l^{(i)}$. That is, assign each sample to the class of the nearest cluster mean.

3. Update the cluster means $\mu_l^{(i+1)}, l = 1, 2, \dots, K$, as the sample mean of all samples in $c_l^{(i)}$

$$\mu_l^{(i+1)} = \frac{1}{N_l} \sum_{\vec{x} \in c_l^{(i)}} \vec{x}, \quad l = 1, 2, 3, \dots, K \quad (5.4)$$

where N_l is the number of samples in $c_l^{(i)}$.

4. If $\mu_l^{(i+1)} = \mu_l^{(i)}$ for all $l = 1, 2, \dots, K$, the algorithm has converged, and the procedure is terminated. Otherwise, go to step 2.

The distance between two feature vectors is measured using a Mahalanobis distance, defined by

$$d(\vec{x}_1, \vec{x}_2) = \sum (\vec{x}_1 - \vec{x}_2)^T S^{-1} (\vec{x}_1 - \vec{x}_2) \quad (5.5)$$

where $\Sigma = \text{diag}(1, 1, 1, 1, 2, 2, 2)$. S^{-1} is the inverse of the variance-covariance matrix of $\vec{x}$. The purpose of the matrix Σ is to weight the color intensities since they are of different scales (and units) from the other 4 components in the feature vector. Here after some trial-and-error, $\Sigma = \text{diag}(1, 1, 1, 1, 2, 2, 2)$ is found to give good results.

The K-means clustering algorithm is thus implemented as follows:

1. Threshold the magnitude of the image velocities (optical flow vectors) to determine approximately the regions of interest (the object's region). The threshold is set to a predefined number (for example 2 pixels/frame)
2. Perform the K-means clustering on the entire region extracted in (1).
3. For each region segmented by the K-means clustering, further segment by connectivity and remove unconnected regions with the size less than a threshold, which is set to a certain number of pixels.

The K-means algorithm normally requires the number of clusters to be known beforehand. Next in this Chapter the K-means cluster validity measures method is described and it overcomes this limitation of having to indicate the number of clusters by incorporating a validity measure based on the intra-cluster and inter-cluster distance measures.

5.2.2 K-means Cluster Validity Measures

Many criteria have been developed for determining the cluster validity, all of which have a common goal to find the clustering which results in compact clusters well separated. The Davies-Bouldin index [170], for example, is a function of the ratio of the sum of the within-cluster scatter to the between-cluster separation. The objective is to minimize this measure as to minimize the within-cluster scatter and maximize the between-cluster separation. Bezdek and Pal [15] have given a generalization of Dunn's index [170]. By considering five different measures of the distance between clusters and three different measures of the cluster diameter, they obtain fifteen different values of the generalized Dunn's index.

Since the K-means method aims to minimize the sum of squared distances from all points to their cluster centers, this should result in compact clusters. The distances of the points from their cluster centers can therefore be used to determine whether the clusters are compact. For this purpose, the intra-cluster distance measure is used, which simply calculates the distance between a point and its cluster center and the average of all of these distances, which is defined as

$$intra = \frac{1}{N} \sum_{i=1}^K \sum_{\vec{x} \in c_i} \|\vec{x} - \mu_i\|^2 \quad (5.6)$$

where N is the number of pixels in the image, K is the number of clusters, and μ_i is the cluster center of cluster c_i . It is obviously necessary to minimize this measure. Meanwhile the inter-cluster distance should be measured, or the distance between clusters, which is required to be as large as possible. The distance between cluster centers is then calculated, and taking the minimum of this value, defined as:

$$inter = \min(\|\mu_i - \mu_j\|^2), \quad i = 1, 2, \dots, K-1 \quad j = i+1, \dots, K \quad (5.7)$$

The minimum of this value is taken as the smallest of this distance to be maximized, and the other larger values will automatically be bigger than this value.

Since both of these measures Eq.5.7 and Eq.5.8 are required to be estimated if there is a good clustering, these two can be combined in some way. The obvious way is to take the ratio, defined as:

$$validity = \frac{intra}{inter} \quad (5.8)$$

Since the intra-cluster distance is to be minimized and this measure is in the numerator, this validity measure is continuously to be minimized. The inter-cluster distance measure is also to be maximized, and since this is in the denominator, the validity measure is again to be minimized. Therefore, the clustering which gives a minimum value for the validity measure will tell what the ideal value of K is in the K-means procedure.

5.2.2.1 Description of the K-means Cluster Validity Measures Method

The K-means cluster validity measures method presented here is taken from [170], which is based on the K-means algorithm. It overcomes the limitation of having to indicate the number of cluster by incorporating a validity measure based on the intra-cluster and inter-cluster distance measures.

The segmented images here are produced with 2 up to K_{max} clusters, where K_{max} is an upper limit on the number of clusters, and then calculate the validity measure to determine which is the best clustering, and therefore, what is the optimal value of K . The first thing is to form one cluster containing all the pixels in the image. Then an iterative process begins where, unless the number of clusters is equal to K_{max} , the cluster having the maximum variance is split into two. Once the cluster is split into two, the K-means procedures are used to obtain the clustering for this new number of clusters. Once all the clusters have been formed, the validity measures can be calculated for each of them to estimate what the optimal value of K is.

Since the K-means algorithm aims to minimize the average intra-cluster distance, it is most likely that the cluster having maximum variances will be separated by the K-means procedure when the number of clusters is increased. Therefore, when the number of clusters is required to be increased, the cluster having the maximum variance is split, so the K-means procedure gives the good starting cluster centers. The variance of the three components for the cluster c_i is calculated as

$$\sigma_{ij}^2 = \frac{1}{N_i} \sum_{\vec{x} \in c_i} (\vec{x} - \mu_{ij})^2 \quad i = 1, 2, \dots, K \quad j = 1, 2, 3 \quad (5.9)$$

where N_i is the number of pixels in the cluster c_i and $\vec{x}$ is the vector representing each pixel's red, green and blue components as x_1 , x_2 , and x_3 , respectively. Here Eq.5.9 is not for the K-means mean estimation but is the biased estimator for the variances of the three components for the cluster c_i . This equation gives the three variance values, but only one value is ultimately wanted, which can be used to compare the variance of each cluster. So the average variance of the three components is taken by adding them

up and dividing the sum by 3. This gives the following variance value

$$\sigma_i^2 = \frac{1}{3} \sum_{j=1}^3 \sigma_{ij}^2 \quad i = 1, 2, \dots, K \quad (5.10)$$

When splitting a cluster, all the Red, Green and Blue components are taken into account. Given the cluster c_i whose cluster center is μ_i , two new cluster centers μ'_i and μ''_i are required to be obtained. The cluster c_i is split by creating two new values for each component, which are centered around the cluster center value. The two new clusters centers are calculated as:

$$\begin{aligned} z'_i &= (\mu_{i1} - a_1, \mu_{i2} - a_2, \mu_{i3} - a_3) \\ z''_i &= (\mu_{i1} + a_1, \mu_{i2} + a_2, \mu_{i3} + a_3) \end{aligned} \quad (5.11)$$

where a_1 , a_2 and a_3 are constants. The values for these constants are estimated by taking into account the minimum and maximum values for each color component occurring in the cluster. The constants a_j will be the values which are half of the smaller of $(\mu_{ij} - \min_j)$ and $(\mu_{ij} - \max_j)$, where $\min_j$ is the minimum value for the j th color component and $\max_j$ is the maximum value for the j th component. These result in the two new clusters centers, which are well separated but also still well within the original cluster.

With the above method any validity measures such as the Davies-Bouldin index or Dunn's index can be used [170]. For Dunn's index the clustering which maximizes this index is just required to be found.

By incorporating the validity measures based on the intra-cluster and inter-cluster distance measures, the number of clusters present in an image can be determined automatically. There is a tendency to select smaller cluster numbers for natural images, which is due to the inter-cluster distance being much greater and greatly affecting the validity measures. This can be overcome by looking for a local maximum for the validity measure and then by finding the minimum value after the local maximum. By using this modified rule, the normal smallest number of clusters that can be selected is 4 [170]. This is not a real problem because natural color images can be expected to have more

than two or three clusters. The modified rule still allows the optimal number of clusters to be selected for the synthetic images, except for the image with only two clusters as two clusters cannot be selected by this modified rule. The other methods such as the Davies-Bouldin index and Dunn's index could not detect the correct number of clusters for all the natural images. However, the validity measure presented in this Chapter performs more consistently for all of the natural images, producing good segmentation results.

As minor modifications to this algorithm, the median cluster center representation can be used instead of the mean cluster center representation, and the absolute distance can be used instead of the Euclidean distance to calculate the distance between a pixel and its cluster center or between the cluster centers. It can also be used in any of the numbers of different color spaces available. As an improvement can be done by incorporating the context of the image, a given pixel is expected to be highly correlated with its neighboring pixels values. This could be achieved by taking into account the values of the neighboring pixels.

The method from [170] is not restricted to the color images. It can be easily extended to cope with any dimensionality, so this method may also be used for multi-spectral images. Similarly, there is no reason why this method cannot be used for grey scale images, which have only one dimension.

5.3 Fusion Based Video Images Segmentation Using Unsupervised Bayesian On-line Learning

5.3.1 Introduction

In this part a novel algorithm is developed for video images clustering and segmentation. Video segmentation is different from a series of single image segmentations (such as the previous K-means image segmentation). Single image segmentation can yield very different results for two very similar images [89]. Video image segmentation demands that for a given image, the segmentation achieved should relate to segmentation of the

previous images, provided that they belong to the same shot. In such a way the clustering function should have the ability to update its classification parameters to adapt to the image visual features' changes.

Here the video images segmentation is developed in a probabilistic framework based on the multiple visual features fusion. In the probability based algorithm, the prior probabilities of each cluster $P(c_i)$ and the cluster-conditional density for each pixel $P(x|c_i)$ should be known. Unfortunately, normally pattern recognition applications rarely have the complete knowledge about the probabilistic structure for the real-time updating. Then the solution to this problem is to use some design samples or training data to estimate these unknown probabilities and probability densities, and then use the resulting estimates as if they are the true values. Here if assuming that $P(x|c_i)$ is a normal density with the mean μ_i and covariance matrix σ_i , then the above assumption simplifies the problem from one of estimating an unknown function $P(x|c_i)$ to one of estimating the parameters μ_i and σ_i [46] [101].

Currently some work have been done on the probability based image segmentation using multiple visual features fusion. Khan et. al. [101]'s approach of combining multiple features for the segmentation involves finding the weights of each feature at a pixel. Khan et. al. use the maximum likelihood estimate for classification. In their work it is assumed that the visual feature's PDF parameters between two consecutive images are the same. However this assumption is not quite accurate when the target's visual features and motions have great or uncertain variations. Li et. al. [117] present a color image segmentation method that aims for the content-based image retrieval systems. Their method combines the region growing and Bayesian learning approaches. The parameters of the distribution for a region are learned by an on-line learning procedure based on a variation of the Bayesian learning. Once the regions have been grown, a merging operation is carried out to merge the small regions into their most similar large neighboring region.

Concluded from the previous work, there are two common and reasonable parameter estimation approaches, namely, maximum-likelihood estimation [46] [101] [224] and Bayesian estimation [46] [117] [86] [200] for the probability-based image segmentation. The re-

sults obtained with these two procedures are frequently nearly identical, but the approaches are conceptually quite different [46]. Maximum-likelihood and several other methods view the PDF parameters as quantities whose values are fixed but unknown. The best estimate of their value is defined to be the one that maximizes the probability of obtaining the samples actually observed. However with the fixed PDF parameters, the maximum-likelihood estimation algorithm is not quite accurate under the situation when the distributions have some great changes between two consecutive images [101]. In contrast, the Bayesian methods view the parameters as random variables with some known prior distributions. Observation of the sample converts this to a posterior density, thereby revising the opinion about the true values of the parameters. In the Bayesian case, a typical effect of observing additional samples is to sharpen the a posteriori density function, causing it to peak near the true values of the parameters. This phenomenon is then known as the Bayesian learning [46] [140]. Compared with Maximum-Likelihood, the Bayesian method is an adaptive and learning-based approach, which updates the parameters' estimates with new samples. In general, the Bayesian method uses more of the information brought to the problem than the maximum-likelihood method. If such information is reliable, the Bayesian methods can be expected to give better results.

In this Chapter the Bayesian estimation method is developed as the unsupervised on-line learning approach for the features clusters' PDF parameters identification. This contributes the proposed algorithm as a novel method for the probability based image pixels clustering. The Bayesian estimation is to use the image's prior clustering information to design a classifier and then use the decisions of the classifier to label the new image pixels, which is also called the decision-directed approach to unsupervised learning. In this Chapter the decision-directed approach is applied sequentially on-line by updating the classifier each time when a new image frame is clustered.

There are many references to the Bayesian estimation method [46], but during the past few years there are too few on using it for video images segmentation. Several recent and representative algorithms using the Bayesian estimation for video images segmentation are reviewed here. Wang et. al. [216] propose a unified probability based framework for spatio-temporal segmentation of video sequences. The spatial and temporal consistency is expressed in terms of interactions between the motion field, the intensity segmentation

field, and the video segmentation field through a Bayesian network. The notions of distance transformation and Markov random field are used to express spatio-temporal constraints. Given consecutive frames, an optimization method is proposed to maximize the conditional probability density of the three fields in an iterative way. In their method in order to simplify the computation, they ignore the localization properties in the image sequences. Moreover, adaptively and automatic determination of the number of layers and selection of the parameters are not considered in their method.

Sista et. al. [99] [190] present another solution to the problems of intensity image segmentation using Bayesian estimation in a multiscale setup. Their approach regards the number of regions, the data partition and the parameter vectors that describe the probability densities of the region as unknown. They compute the MAP estimates jointly by maximizing their joint posterior probability density given the data. Since the estimation of the number of regions is also included into the Bayesian formulation, they have a fully automatic or unsupervised method of segmenting images. They provide a descent algorithm that starts with an arbitrary initial segmentation of the image when the number of regions is known and iteratively computes the MAP estimates of the data partition and associated parameter vectors of the probability densities. However the descent algorithm is an iterative estimation method, which requires much more computational time for it to converge.

Jojic et. al. [92] propose a technique for automatically learning layers of “flexible sprites”-probabilistic 2D appearance maps and masks of moving, occluding objects. The model explains each input image as a layered composition of flexible sprites. A variational Expectation Maximization (EM) algorithm is used to learn a mixture sprites from a video sequence. For each input image, the Bayesian probabilistic inference is used to infer the sprite class, translation, mask values and pixel intensities. The EM algorithm is an iteratively estimation method, which also requires more computational time. Learning by the EM algorithm in a frame by frame rate is not suitable for the real-time on-line video images segmentation.

Different from the above methods, the algorithm proposed in this Chapter is more simple and faster to implement. Different visual features are easily combined through the

Bayesian probability theory with a weighting algorithm. No Bayesian network nor Markov random field is required. Also the method in this Chapter adaptively learns and updates the classification function's PDF parameters through the image sequence. At the same time the image localization information is included into the classification function by using the spatial distribution function. The learning process in the proposed algorithm just uses the previous classification information to update the current parameters. In such a way no iterative estimation methods such as the descent or EM algorithms are required, which thus greatly reduce the computational time. All these have made the proposed method a better solution for the probability based video images pixels clustering.

The proposed unsupervised Bayesian on-line learning based image pixels clustering procedures are as follows: Multiple visual features are first extracted from the video images. Each feature's cluster is assigned with a Gaussian Distribution Function. They are fused together in a Bayesian probability framework with a weighting algorithm. Then the image clustering is carried out by assigning each image pixel the cluster number with the maximum value from the posterior likelihood. The first image from the video file is segmented using the EM algorithm as the feature clusters' PDF parameters initialization. Then with the previous image features clusters' PDF parameters and the image pixels classification information, the unsupervised Bayesian on-line learning approach is developed to update the features clusters' PDF parameters for the next image clustering. Using this on-line learning process, the segmentation algorithm can handle the problems such as the large or uncertain object's visual and dynamic motion variations or the object occlusion problems in the image sequence.

5.3.2 Probability Based Multiple Visual Features Fusion Segmentation Theory

5.3.2.1 Different Image Visual Features

In this part different image visual features are assigned with a probability distribution function (PDF). First, for every pixel $I_t(x, y)$ in the image I_t (t denotes the time index)

there is a 7-dimensional feature vector for the pixels clustering, which is given by:

$$\vec{x} = [x, y, R, G, B, v_x, v_y] \quad (5.12)$$

where (x, y) denote the spatial locations, $[R, G, B]$ denote the 3D color components and $[v_x, v_y]$ denote the two-dimensional optical flow vectors at the pixel. All these values can be measured separately at each pixel. Here the spatial location and spread of each segment is used as a feature to make least changes in location of segments. The location (x, y) and the color $[R, G, B]$ are given directly by reading the image file. In this method the optical flow component $[v_x, v_y]$ is computed using the modified Horn and Schunck's method [71].

Assuming the image sequence is segmented into K clusters (classes), denoted by c_i ($i = 1$ to K). Then first the Gaussian PDF is assigned for the clusters of color and optical flow components. The idea of using Gaussian PDFs here is based on the algorithms developed by Khan et. al. [101]. The Gaussian distribution is normally used to indicate the distributed noise around the color or flow mean of a class, which is suitable for the probability based image pixels clustering.

$$P_{c,m}(\vec{x}_t(x, y)|c_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{(\vec{x}_t - \mu_i)^T \Sigma_i^{-1} (\vec{x}_t - \mu_i)}{2}} \quad (5.13)$$

In this equation, d is 2 for the two dimensional optical flow vectors PDFs. 3 is for the three color components R , G and B PDFs. In Eq.(5.13) μ_i and Σ_i refer to the mean and covariance matrix of the feature vectors belonging to the segment c_i .

For the spatial component however, a Gaussian PDF is not quite reasonable. Segments in the image can be of any arbitrary shape and not necessarily elliptical. Thus the PDF is estimated from the actual data, rather than a parametric model (Khan et. al. [101]):

$$P_s(X) = \frac{m^{-1}}{(2\pi)^{d/2} \sigma_s^d} \sum_{i=1}^m \exp \left\{ -\frac{(X - X_{ki})^T (X - X_{ki})}{2\sigma_s^2} \right\} \quad (5.14)$$

where X_{ki} is the i th data point from the cluster k , m is the total number of data points, σ_s is a smoothing parameter and d is the dimensionality of the space (in the case of

spatial PDF it is 2). This PDF (Eq.(5.19)) is generated using the method from [101]. A Gaussian distribution is first generated for each point in the image cluster. Each point is the mean of its own Gaussian distribution. Then all these distributions are added together in Eq.5.19 to form a distribution for the spatial component of this image cluster. This PDF can be efficiently implemented by convolving the binary mask of a class with a Gaussian kernel. The smoothing parameter σ_s governs the width and smoothness of the resulting PDF. The values of σ_s effectively limit the amount of motion allowed between two consecutive frames, and if a large movement is expected within two frames (due to fast objects close to the camera or low frame-rate) a larger σ_s should be used.

5.3.2.2 Multiple Visual Features Fusion for Probability based Video Images Segmentation

After each visual feature's PDF has been decided, this part proposes the algorithm to fuse them for the image clustering and segmentation. The segmentation problem can be expressed in the Bayesian framework as [40]. First let the cluster assignments of each pixel in I_t be denoted by l_t , i.e. for all pixels $I_t(x, y)$, $1 \leq l_t(x, y) \leq K$. Let the probability of a pixel (x, y) in the image I_t belonging to cluster c_i ($1 \leq i \leq K$) be given by $P(c_i|\vec{x}_t(x, y))$, where $\vec{x}_t(x, y)$ is the feature vector of pixel (x, y) at time t .

$$P(c_i|\vec{x}_t(x, y)) = \frac{P(\vec{x}_t(x, y)|c_i)P(c_i)}{P(\vec{x}_t(x, y))} \quad (5.15)$$

This pixel is assigned the cluster number with the maximum probability as shown in:

$$l_t(x, y) = \arg \max_i \left\{ \frac{P(\vec{x}_t(x, y)|c_i)P(c_i)}{P(\vec{x}_t(x, y))} \right\} \quad (5.16)$$

where $1 \leq i \leq K$. The denominator can be ignored because it does not depend on i and is always positive. Then with a log function, the likelihood relationship is:

$$l_t(x, y) = \arg \max_i \{ \ln(P(\vec{x}_t(x, y)|c_i)) + \ln(P(c_i)) \} \quad (5.17)$$

Here the prior term $\ln(P(c_i))$ in Eq.(5.17) has the same value for each cluster since all clusters are equally likely to be observed. So the prior $\ln(P(c_i))$ can be considered not affecting the image segmentation and clustering and omitted in the following equations.

Since the spatial, color and optical flow terms are assumed to be the mutually independent visual features and estimated separately, the probability $P(\vec{x}_t(x, y)|c_i)$ can be broken down into a product of three probability terms. This assumption is valid because given the spatial location of a segment, its color or optical flow vectors can not be predicted. Similar, a particular color may occur anywhere in the image and that segment might be moving with any velocity. Since the space, color and flow terms are considered separately, individual weights ω can be assigned to each of the terms, if one of the features needs to be emphasized more than the others. Then the log-likelihood equation can be written as Eq.(5.18)

$$l_t(x, y) = \arg \max_i \{ \omega_t^s(x, y) \ln P_s(X) + \omega_t^c(x, y) \ln P(x_t^c(x, y)|c_i) + \omega_t^f(x, y) \ln P(x_t^f(x, y)|c_i) \} \quad (5.18)$$

where s , f and c denote space, optical flow and color components.

In order to get the maximum value of Eq.(5.18), each feature's cluster-condition function $P(\vec{x}_t(x, y)|c_i)$ should be known with every frame. Here the first frame of each shot from the video file is segmented using some clustering scheme which is called the initial image segmentation. Then the segments of the image sequence are computed iteratively using the learned features clusters' PDF parameters.

5.3.2.3 Weighting Algorithm for Probability based Multiple Visual Features Fusion

Individual weights are used for each set of features for each pixel in frame (Eq.(5.18)). The weights can be computed based on the methodologies developed by [101].

The motion and color components are the factors affecting the tracking mostly. If the flow is 'reliable', then the flow term is weighted more, and correspondingly decrease the weight of the color term, and vice versa. Thus the weights here are adaptively estimated based on the motion estimation. Here the value of the optical flow likelihood with each

5.3 Fusion Based Video Images Segmentation Using Unsupervised Bayesian On-line Learning

91

cluster $\ln P(x_t^f(x, y)|c_i)$ is observed to estimate the weights. The higher the probability $\ln P(x_t^f(x, y)|c_i)$ is, the better this pixel matches to one of the existing clusters. If the probability $\ln P(x_t^f(x, y)|c_i)$ is low, it indicates that a higher weight should be given to the color term in those areas. Secondly, if an object moves with respect to the background but then stops, it should not be immediately made the part of the background, but should still be segmented out. This effect is catered by computing the absolute difference between the maximum value of $\ln P(x_t^f(x, y)|c_i)$ and the second maximum value from amongst the significant neighbors of this cluster. If the difference in their probabilities is low, then this means that the optical flow is not providing enough discriminatory information. In such a scenario, the color component should also be weighted more. Next the details about how to automatically obtain the weights are explained.

Based on the above explanations and from the formulas developed in [101], the log likelihood $L_t^f(x, y, i)$ can be normalized between 0 – 1 by multiplying it with a sigmoid function. This yields the confidence measure ρ_1 .

$$\rho_1 = \{1 + \exp(-a(\max_i(L_t^f(x, y, i))))\} \quad (5.19)$$

where a is a positive number defining the slope of the sigmoid function. Here $a = 0.5$ is used in all the experiments. Then

$$d = |\max(L_t^f(x, y)) - \max_2(L_t^f(x, y))| \quad (5.20)$$

where $\max_2$ is the highest probability value returned from the neighbors of the cluster which returned the highest probability value. Since d is always positive, it is normalized between 0-1 by multiplying it with a shifted sigmoid function

$$\rho_2 = \{1 + \exp(-a(d - t))\} \quad (5.21)$$

where t is an appropriate shift parameter for the sigmoid function.

Having computed ρ_1 and ρ_2 , the weights ω^f (optical flow component) and ω^c (color

component) as compliments of each other can be computed automatically next:

$$\omega^f = \rho_1 \rho_2 \quad (5.22)$$

$$\omega^c = 1 - (\rho_1 \rho_2) \quad (5.23)$$

5.3.3 Initial Image Segmentation Process

This part describes the algorithm for the initial image frame segmentation. The initial image segmentation is to accurately generate the classification function's PDF parameters. Some methods like the K-means algorithm proposed previously can be used here. However disregarding the computational cost, in this thesis the recursive method—the “EM” (Expectation-Maximization) algorithm is selected. The EM algorithm is a well-known and accurate parameter estimation methods, which can give satisfactory initial estimates for the unsupervised Bayesian on-line learning based image segmentation [86] [101]. The K-means algorithm is simpler and faster, but it is not an accurate method to estimate the initial parameters.

The first step of the initial image segmentation is to apply the EM algorithm on the color, spatial geometric and optical flow data of the image to essentially find the Gaussian distribution functions that fit the $7D$ image pixel vector (Eq.(5.12)). The central equation of the EM algorithm in this thesis is developed from [46]. If let i be an iteration counter, then the parameter vector θ^i is the current (best) estimate for the assigned clustered feature distribution; θ is a candidate vector for an improved estimate for the next step. Given such a candidate θ , the likelihood of the data $P(\theta; \theta^i)$, including the image feature points D marginalized with respect to the current best distribution (θ^i), is estimated. Different candidates θ s will lead to different such likelihoods. The algorithm will select the best such candidate and call it θ^{i+1} —the one corresponding to the greatest $P(\theta; \theta^i)$. Now if let T_1 be a preset convergence criterion, the EM algorithm is as follows:

1. begin initialization $\theta^0, T, i = 0$
2. do $i \leftarrow i + 1$

3. E STEP: compute $P(\theta; \theta^i)$
4. M STEP: $\theta^{i+1} \leftarrow \arg \max_{\theta} P(\theta; \theta^i)$
5. until $P(\theta^{i+1}; \theta^i) - P(\theta^i; \theta^{i-1}) \leq T$
6. return $\hat{\theta} \leftarrow \theta^{i+1}$
7. end

Next it is to take the regions generated by the first step and use them to compute clusters in the likelihood equation to compute the new regions. This process is repeated a number of times for the same frame, until the clustered regions are stable. The clusters thus obtained are suitable to be used for the next frame clustering.

5.3.4 Unsupervised Bayesian On-line Learning for Gaussian Distribution Case

Since the target's visual feature clusters' PDFs are assigned to be the Gaussian PDF, the unsupervised Bayesian on-line learning algorithm is developed next for the Gaussian distribution parameters estimation. The learning is based on the previous image only not on the complete set of the already computed images from the video sequence. But since the updating is carried out at a frame by frame rate, all the previous image classification functions and parameters will slowly adapt to the current images' visual features. In this section the step by step parameters estimation is explained in details.

Here the learning process for each visual feature cluster's Gaussian PDF parameters is the same, so x is used to represent the image pixel in the clustering function instead of using the image pixel's 7-dimensional feature vector $\vec{x}$ from Eq.(5.12).

Let D denote the set of pixels in the previous segmented frame, then the Bayesian classifier Eq.(5.15) becomes

$$P(c_i|x, D) = \frac{P(x|c_i, D)P(c_i|D)}{\sum_{j=1}^c p(x|c_j, D)P(c_j|D)} \quad (5.24)$$

As this equation suggests, the information provided by the previous clustered image pixels can be used to help estimate both the cluster-conditional densities and the prior probabilities of the classifier for the next frame. Here the data points can be separated by clustering into K subsets $D_1, \dots, D_K$, with the data in D_i belonging to cluster c_i . Then when addressing maximum-likelihood methods for Eq.(5.17), the data points in D_i have no influence on $P(x|c_j, D_j)$ if $i \neq j$. There are two simplifying consequences. First it allows the classification algorithm to work with each cluster separately, using only the samples in D_i to estimate $P(x|c_i, D_i)$. Then Eq.(5.24) can be written as

$$P(c_i|x, D_i) = \frac{P(x|c_i, D_i)P(c_i)}{\sum_{j=1}^c P(x|c_j, D_j)P(c_j)} \quad (5.25)$$

Secondly, because each cluster can be treated independently, needless cluster distinctions can then be dispensed to simplify the notation.

Although the desired probability density $P(x|c_i, D_i)$ for the next image frame clustering is unknown, it can be assumed that it has a known parametric form. The only thing assumed unknown is the value of a parameter vector θ_i for this known parametric form. Any information about θ_i prior to observing the samples is assumed to be contained in a known prior density $P(\theta_i)$. Observation of the samples converts this to a posterior density $P(\theta_i|D_i)$, which is sharply peaked about the true value of θ_i . Because the selection of x_i and that of the previous pixels in D_i is done independently, thus

$$P(x_i|c_i, D_i) = \int P(x_i, \theta_i|c_i, D_i)d\theta = \int P(x_i|c_i, \theta_i)P(\theta_i|c_i, D_i)d\theta \quad (5.26)$$

This equation links the desired cluster-conditional density $P(x_i|c_i, D_i)$ to the posterior density $P(\theta_i|c_i, D_i)$ for the unknown parameters.

Here each image cluster is equally estimated thus c_i and i can be omitted in the following formulas.

To indicate explicitly the number of previous segmented image frames in a set for a single category, $E^n = \{I_1, \dots, I_n\}$. Then with the image's independence assumption

$$P(E^n|\theta) = \prod_{m=1}^n P(I_m|\theta) \quad (5.27)$$

it can be obtained

$$\begin{aligned} P(E^n|\theta) &= P(I_n|\theta)P(E^{n-1}|\theta) \\ P(\theta|E^n) &= \frac{P(I_n|\theta)P(\theta|E^{n-1})}{\int P(I_n|\theta)P(\theta|E^{n-1})d\theta} \end{aligned} \quad (5.28)$$

Through the above equations it is called the recursive Bayesian approach to the parameter estimation for the image segmentation. This is an incremental or on-line learning method, by which the learning goes on as the data are collected.

5.3.4.1 The Unsupervised Bayesian On-line Learning Algorithm

In this thesis Bayesian estimation techniques for the probability based image segmentation are developed to learn the features clusters' PDF parameters with the previous classified image information. The learning algorithm calculates the a posteriori density $P(\theta|D)$ and the desired probability density $P(x|D)$ for the case where $P(x|\theta) \sim N(\theta, \sigma)$.

Following assumptions have been first made:

1. The number K of classes is known.
2. The prior probabilities $P(c_j)$ for each class are known, $j = 1, \dots, K$.
3. The forms for the class-conditional probability density $P(x|c_j, \theta_j)$ are known, $j = 1, \dots, K$, but the full parameter vector $\theta = (\theta_1, \dots, \theta_K)$ is not known.
4. Part of the knowledge about θ is contained in a known prior density $P(\theta)$.
5. The rest of the knowledge about θ is contained in a set D of n samples $x_1, \dots, x_n$ drawn independently from the familiar mixture density

$$P(x|\theta) = \sum_{j=1}^n P(x|c_j, \theta_j)P(c_j) \quad (5.29)$$

Based on the above assumptions first the image frame I_t segmentation is based on the priori distribution parameters μ_{t-1}, σ_{t-1} from the frame I_{t-1} . Then with the clustered

5.3 Fusion Based Video Images Segmentation Using Unsupervised Bayesian On-line Learning

96

n pixels information of the frame I_t , the Bayesian iterative learning algorithm is used to estimate the parameters μ_t and σ_t for the next frame I_{t+1} segmentation.

Here using the Bayesian learning to estimate the covariance and mean simultaneously is difficult since they are coupled with each other [46] [117]. Since the variance is often at the same order of the magnitude, compared with the mean θ , the variance σ is a less effective factor for the probability based image classifier. Thus for the I_{t+1} frame segmentation, it is reasonable to assume that only the mean θ is unknown. The prior knowledge about μ_t is then expressed by a known prior density $p(\theta)$ which is

$$P(\theta) \sim N(\theta_0, \sigma_0^2) \quad (5.30)$$

where both the θ_0 and σ_0 are known. θ_0 represents the best prior guess for the μ_t and σ_0 measures the uncertainty about this guess. Here the assumption that the prior distribution for θ is normal will simplify the subsequent mathematics [46].

Having selected the prior density for θ , let $D = \{x_1, \dots, x_n\}$ from the previous clustered pixels in the frame I_{t-1} then

$$\begin{aligned} P(\theta|D) &= \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta} \\ &= \alpha \prod_{m=1}^n P(x_m|\theta)P(\theta) \end{aligned} \quad (5.31)$$

where α is a normalization factor that depends on D but is independent of θ . Because of $P(x_m|\theta) \sim N(\theta, \sigma)$ and $P(\theta) \sim N(\theta_0, \sigma_0^2)$, it is shown that

$$\begin{aligned} P(\theta|D) &= \alpha \prod_{m=1}^n \underbrace{\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_m - \theta}{\sigma}\right)^2\right]}_{P(x_m|\theta)} \underbrace{\frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\theta - \theta_0}{\sigma_0}\right)^2\right]}_{P(\theta)} \\ &= \alpha' \exp\left[-\frac{1}{2}\left(\sum_{m=1}^n \left(\frac{\theta - x_m}{\sigma}\right)^2 + \left(\frac{\theta - \theta_0}{\sigma_0}\right)^2\right)\right] \\ &= \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)\theta^2 - 2\left(\frac{1}{\sigma^2} \sum_{m=1}^n x_m + \frac{\theta_0}{\sigma_0^2}\right)\theta\right]\right] \end{aligned} \quad (5.32)$$

where factors that do not depend on θ have been absorbed into the constants α , α' and

5.3 Fusion Based Video Images Segmentation Using Unsupervised Bayesian On-line Learning

97

α'' . Thus, $P(\theta|D)$ is an exponential function of a quadratic function of θ . Because this is true for any number of previous clustered image pixels, $P(\theta|D)$ remains normal as the number of n of samples is increasing, and $P(\theta|D)$ is said to be a reproducing density and $P(\theta)$ is said to be a conjugate prior [46]. Then $P(\theta|D)$ can be written as $N(\theta_n, \sigma_n)$. θ_n and σ_n can be found by equating coefficients in Eq.(5.32) with corresponding coefficients in the generic Gaussian of the form

$$P(\theta|D) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\theta - \theta_n}{\sigma_n}\right)^2\right] \quad (5.33)$$

With the $\hat{\theta}_n$ is the n sample mean

$$\hat{\theta}_n = \frac{1}{n} \sum_{m=1}^n x_m \quad (5.34)$$

it is shown that

$$\theta_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right)\hat{\theta}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\theta_0 \quad (5.35)$$

and

$$\sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2 + \sigma^2} \quad (5.36)$$

Here σ is the previous covariance matrix σ_{t-1} . These equations (Eq.(5.35) and Eq.(5.36)) show how the prior PDF information (θ_0, σ_0) is combined with the empirical information $(\hat{\theta}_n, \sigma_{t-1})$ in the current segmented frame to obtain the *a posteriori* density $p(\theta|D)$. Roughly speaking, θ_n represents the best guess for θ after observing n clustered pixels from the image frame I_t , and σ_n^2 measures the uncertainty about this guess.

Having obtained the *a posteriori* density for the mean, $P(\theta|D)$, all that remains is to obtain the “cluster-conditional” density $P(x|D)$ for the I_{t+1} image frame clustering. From Eq.(5.26), (5.30) and (5.33) it is shown

$$\begin{aligned} P(x_t|D) &= \int P(x_t|\theta)P(\theta|D)d\theta \\ &= \int \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_t - \theta}{\sigma}\right)^2\right] \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{1}{2}\left(\frac{\theta - \theta_n}{\sigma_n}\right)^2\right] d\theta \\ &= \frac{1}{2\pi\sigma\sigma_n} \exp\left[-\frac{1}{2}\frac{(x_t - \theta_n)^2}{\sigma^2 + \sigma_n^2}\right] f(\sigma, \sigma_n) \end{aligned} \quad (5.37)$$

That is, as a function of x_t , $P(x_t|D)$ is proportional to $\exp[-(1/2)(x_t - \theta)^2/(\sigma^2 + \sigma_n^2)]$, and hence $P(x_t|D)$ is normally distributed with the mean θ_n and variance $\sigma^2 + \sigma_n^2$

$$P(x_t|D) \sim N(\theta_n, \sigma^2 + \sigma_n^2) \quad (5.38)$$

So to obtain the cluster-conditional density $P(x_t|D)$, whose parametric form is known to be $P(x_t|\mu) \sim N(\mu_t, \sigma_t^2)$, then

$$\begin{aligned} \mu_t = \theta_n &\rightarrow \mu_{j,t} = \theta_{j,n} \quad 1 \leq j \leq K \\ \sigma_t^2 = \sigma_{t-1}^2 + \sigma_n^2 &\rightarrow \sigma_{j,t}^2 = \sigma_{j,t-1}^2 + \sigma_{j,n}^2 \quad 1 \leq j \leq K \end{aligned} \quad (5.39)$$

where θ_n can be obtained from Eq.(5.35) and Eq.(5.36), and σ_n^2 can be obtained from Eq.(5.36). Here j is one of the cluster numbers in the image.

This is the final result for the PDF parameters learning: for the feature cluster j the density $P(x_t|D)$ is the desired cluster-conditional density $P(x_t|c_j, D_j)$, then together with the prior probabilities $P(c_j)$ it gives the probabilistic information needed in Eq.(5.17) and Eq.(5.18) to do the clustering for the I_{t+1} image frame. This is in contrast to the maximum-likelihood methods that only make the previous frame point estimates for $\mu_{j,t}$ and $\sigma_{j,t}$, rather than estimate a distribution for $p(x|D)$ in real-time learning.

5.4 Experimental Results

To evaluate the performance and adjust the parameters of the proposed methods, the clustering and segmentation algorithms have been tested on several real image sequences. The video image sequences used in these testings are about 30 – 50 frames long.

5.4.1 K-means Image Segmentation Results

First, the K-means algorithm is tested on the normal moving target and moving background image sequence. The image size is 256×220 pixels.

From Fig.(5.1) a blue target is moving in the image sequence in a cluttered background.

Figure 5.1: Moving object with moving background in the indoor situation

As it can be seen directly from the images, the background is changing frame by frame.

The K-means segmentation and clustering algorithm is implemented to segment the image sequence into several regions as shown in Fig.(5.2). The K-means clustering

Figure 5.2: The motion vectors field segmentation and clustering results from the image thresholding and the K-means clustering.

algorithm is sensitive to slight movements in the images, such as movements due to wind. Therefore, it is necessary to eliminate these vectors due to wind etc. by thresholding.

Fig. (5.2.1), Fig. (5.2.2) and Fig. (5.2.3) show the motion regions after the thresholding. The motion vectors smaller than this threshold are removed from the optical flow vectors field. Next the clusters with smaller areas are removed with the use of another threshold. In this way only the significant movements in the image sequence are extracted. This process can be explained in a few steps as follows:

1. All the image pixels are labelled according to their assigned cluster numbers. The labelling indicates the connected motion regions from the clustered images.
2. The size of each region is compared with a preset threshold. If the region size is smaller than this size threshold, it will be removed from the clustered images.
3. Once all the small regions are removed, the “close” morphological operations (dilation followed by erosion) are implemented on the clustered images. The “close” morphological operations [171] smoothen the contours of the segment. That is, it eliminates small holes, and fills gaps in the contour.

Fig.(5.2.4), Fig.(5.2.5) and Fig.(5.2.6) show the K-means clustering results. Here, different colors represent different movements. From the results it can be seen that the region of the target has the same color within one image, which makes the identification of the region of interest (ROI) an easy task. However it can be also seen that the same object’s regions in different images have different colors, thus the object’s assigned cluster number is changing in different frames. This reflects that the K-means algorithm is not very robust to update the classifier functions when the target and background’s visual features are changing. The object can not be classified into the same cluster using the K-means algorithm.

5.4.2 Unsupervised Bayesian On-line Learning based Video Images Segmentation Results

Different from the K-means algorithm, in the unsupervised Bayesian on-line learning based image segmentation the EM iteration algorithm is first used for the image’s initial segmentation (the same image sequence of Fig.(5.1)).

With the EM algorithm, the visual feature clusters' Gaussian PDF parameters are initialized.

Fig.(5.3) and Fig.(5.4) show the PDF parameters initialization process (PDF's mean and variance) from the EM iteration process. Five different colors represent five different image clusters. Several initial values are first assigned to the features clusters' PDF parameters. With the EM algorithm, the clusters' PDF parameters converge to some values after certain times of iterations. The final converged values are used as the initialized visual features clusters' PDF parameters for the next image frame segmentation.

In Fig.(5.3) the cameras and objects are both moving, so sometimes the object has no movement in the images. In such a situation, the optical flow vectors field only has the background motion. Normally the motion based segmentation algorithm like [89] can not accurately cluster and segment such stationary objects from the image sequence. However in the algorithm proposed in this Chapter, with the adaptive visual features weighting algorithm, the target's color components are weighted more when the object stops moving. And with the unsupervised Bayesian on-line learning, the visual features clusters' PDF parameters are updated. In this way the object can still be clustered out from the image sequence (Fig.(5.6)).

After initializing the visual features clusters' PDF parameters, the probability based algorithms with the unsupervised Bayesian on-line learning is implemented to segment and cluster the image sequence.

First Fig.(5.5) shows the image clustering and segmentation results without the Bayesian on-line learning. The image features' clusters PDF parameters are estimated from the Maximum-Likelihood algorithm. From the images, when the blue object moves close to the cameras, its visual features have some variations (the object's size looks bigger and bigger from left to right.). If without learning, the image clustering and segmentation results are not good. The blue object in the clustered images is not within one unique color and its shape is not accurate.

Fig.(5.6) shows the image clustering and segmentation results with the unsupervised Bayesian on-line learning. Different colors represent different image clusters. The blue moving object in the clustered images has the same color through the whole image

5.3.1:

5.3.2:

5.3.3:

5.3.4:

5.3.5:

Figure 5.3: The image clusters' PDF parameters initialization (mean) with the EM algorithm for the Fig.(5.1) segmentation.

5.4 Experimental Results

103

5.4.1:

5.4.2:

5.4.3:

5.4.4:

5.4.5:

Figure 5.4: The image clusters' PDF parameters initialization (variance) with the EM algorithm for the Fig.(5.1) segmentation.

Figure 5.5: The image clustering and segmentation results without the Bayesian learning. The results shown here correspond to the original figures in Fig.5.1.

Figure 5.6: The image clustering and segmentation results with the unsupervised Bayesian on-line learning (5 Clusters). The results shown here correspond to the original figures in Fig.5.1.

sequence. The boundary of the blue target is also quite accurate to reflect the shape of the object. This reflects that the image clustering and segmentation is very accurate.

Here the color of the object's region is the same as some part of the background colors in the clustered image. This is because five clusters are implemented for the clustering algorithm. The background is in a cluttered environment. Some background parts are clustered into the object's cluster, which makes them have the same color. However

from the image clustering results, the object can be easily distinguished from its neighbor areas and it has the same color within its body through the whole images sequence to reflect its shape.

Figure 5.7: The image segmentation results with the unsupervised Bayesian on-line learning. Only one cluster from the clustered images is shown in these figures. The results shown here correspond to the original figures in Fig.5.1.

Fig.(5.7) shows one cluster of the segmentation results. These images have only one color. Here the blue target is segmented out into the same image cluster through the whole image sequence. The shape of the target is also quite accurate.

Figure 5.8: Image clustering and segmentation results with 3 clusters. The results shown here correspond to the original figures in Fig.5.1.

If let the image cluster number be 3, Fig.(5.8) show that the image clustering and segmentation results are also very accurate.

Compared with the previous K-means image clustering results, the region of the target in the clustered images from the learning algorithm has the same color through the whole

image sequence. This reflects that although the image visual features are changing continuously, the target is always accurately clustered into the same image cluster. This also reflects the learning and updating results of the features clusters' PDF parameters through the images sequence.

In this thesis, 5 and 3 clusters are generally set to test the clustering algorithm. These values are selected based on the previous discussion on how to automatically select the image cluster's number for video images sequence segmentation. From the previous K-means validity measures method in [170], 3 – 5 image clusters are generally better for the natural video images sequence segmentation. Also the number of the clusters in the experiments can be adaptively set to the contents of the video frames. For example a small number of clusters for the object based image segmentation of the frames with 10 moving objects are too ambiguous, even though the object of interest can be well covered. So when the background is very cluttered and there are more objects in the image, 8 – 9 clusters would be more appropriate. But if only one or two objects are moving in the image scene, 3 – 5 image clusters can well do the work.

From the experimental testing, the computational time for 5 clusters are larger than 3 clusters. With less clusters the computational time will decrease. But with less image clusters, the clustering and segmentation results will be less accurate. With the unsupervised Bayesian on-line learning the computational time is nearly the same as the one without learning and there are small differences between them. The learning does not put any more computational burden. In the proposed method the computational time for one frame using Matlab (P4-2.4G and 1G Memory PC) is about 10 seconds. Although Matlab programming is very time consuming, with 5 clusters the computational time is only reasonable. Thus it can be seen that the image cluster's number can not be too large. If the algorithm is implemented using C language, the image segmentation will be a little faster.

The following figures (Fig.(5.9)) show the features clusters' PDF parameters (Mean and Variance) learning process.

In the figures (Fig.5.9 and Fig.5.9), the “-*” lines are the image features clusters PDF parameters with the unsupervised Bayesian on-line learning. The “-o” lines are the

5.9.1:

5.9.2:

5.9.3:

5.9.4:

5.9.5:

Figure 5.9: The image visual features clusters' parameters learning (Mean) for the Fig.(5.1) segmentation.

5.10.1:

5.10.2:

5.10.3:

5.10.4:

5.10.5:

Figure 5.10: The image visual features clusters' parameters learning (Variance) for the Fig.(5.1) segmentation.

parameters from the Maximum Likelihood algorithm without learning (the algorithm from [101]). In each figure five different colors mean five different image clusters. From the calculation with the unsupervised Bayesian on-line learning, the clusters' PDF parameters (mean and variance) have little variation, which is about 10%. Without learning the parameters diverge fast and have large variations, which is up to 40 – 50%. Also it shows that without learning the features clusters may be conflicted with other clusters' parameters. In this way, the image clustering and segmentation has large error, which is not able to segment out the object with its accurate shape as shown in Fig.(5.5).

Figure 5.11: The image sequence with a car driving along the road. The background is changing continuously.

In Fig.(5.11) a car drives along the road with the image view from the front window. As it can be seen directly from the images, the background is changing frame by frame.

Figure 5.12: The car driving image sequence with 3 clusters segmentation results. The results shown here correspond to the original figures in Fig.5.11.

The unsupervised Bayesian on-line learning based clustering and segmentation is implemented with 3 clusters for this car driving image sequence. From the above images, the background buildings and the landmarks along the road are accurately clustered and segmented out from the images. Especially for the yellow car moving along the horizontal direction in the front, it is very accurately segmented out with its exact car shape. For the last few images, the small car almost has no movement, but it is still

accurately segmented out (Fig.(5.12.5)).

Two people are moving towards each other in an outdoor cluttered environment (Fig.(5.13)).

Figure 5.13: Two persons are moving towards each other.

In such a cluttered environment the unsupervised Bayesian on-line learning based algorithm gives accurate video image segmentation.

Figure 5.14: The two persons moving image sequence segmentation results without the unsupervised Bayesian on-line learning. The results shown here correspond to the original figures in Fig.5.13.

Figure 5.15: The two persons moving image sequence segmentation results with the unsupervised Bayesian on-line learning. The results shown here correspond to the original figures in Fig.5.13.

From the above figures (Fig.(5.14) and Fig.(5.15)) the segmentation results with the unsupervised Bayesian on-line learning are much better than those without learning. This is because with the learning, the person's visual features clusters' PDF parameters are adjusted frame by frame. Although in an outdoor very cluttered environment, the different features' clusters PDF parameters are updated to converge to the values for the major image clusters, which results in robust images segmentation. Here, since the moving persons have no regular shapes, the segmentation results can only reflect the rough body shapes of the two moving persons.

In the figures (Fig.(5.15)), using the optical flow information alone may also give good results since the background appears to be more or less fixed. The moving persons' apparent motion can be easily segmented out from the background motion field. This situation is already considered in the previous weighting algorithm. Here the optical flow term can be weighted more than the others and become the dominant one, then it is just like only using optical flow vectors to conduct the image segmentation and clustering.

With the unsupervised Bayesian on-line learning, as the target leaves the image view (Fig.(5.16)), the proposed algorithm can make the accurate object segmentation. The

Figure 5.16: A blue object moving out of the image scene.

Figure 5.17: The image segmentation results with the unsupervised Bayesian on-line learning when the object leaves the image view. The results shown here correspond to the original figures in Fig.5.16.

object's region in the images are always with the black color, which reflects that although the object's shape has some changes it can still be segmented out and clustered into the same image cluster in the image sequence.

Fig.5.18 shows that the blue object is occluded by another black color object. Fig.5.19 shows the segmentation results. Here the segmentation algorithm makes an accurate segmentation with accurate shapes of the two objects when they are occluded by the

Figure 5.18: The object is occluded by another object.

Figure 5.19: The image segmentation results with the unsupervised Bayesian on-line learning when the object is occluded by the other object. The results shown here correspond to the original figures in Fig.5.18.

other. Each object is clustered into its cluster constantly through the whole image sequence.

In all the experimental results presented in this section, the object's region in the clustered images has the same color throughout the whole image sequence. Although the image visual features are changing continuously, the target is still accurately clustered into the same cluster. This reflects the learning and updating results of the cluster's

PDF parameters, which make the video image segmentation algorithm adapt to these visual changes. As it can be seen from the above results, this also gives the better image segmentation and clustering performance than the recent methods such as [46] [89] [101].

Some other situations like after the initial image segmentation is done, if another object enters the image scene, the image segmentation and clustering using the conventional methods like [89] [101] will have problems because they can not generate a new cluster or change the existed cluster to fit the new object. However using the unsupervised Bayesian on-line learning based image segmentation and clustering, the image clusters' parameters are adaptively learned and updated frame by frame to adapt to the changes of the image scene. In such a way after some frames, the newly entered object can be accurately clustered out from the images.

5.5 Summary

In this Chapter in order to make the accurate object-based video images segmentation, different image segmentation algorithms are developed and tested in this Chapter.

First, the K-means method is developed for the fusion based image segmentation. From the above experimental testing, it can be seen that the K-means algorithm is a simple and efficient method for clustering the images. However, the biggest challenge in the K-means clustering procedure is the determination of the correct number of clusters, which is assumed to be known. In this thesis the value of K is determined by a simple validity measures method based on the intra-cluster and inter-cluster distance which allows the number of clusters to be determined automatically [170]. Based on this method, the cluster number for natural image sequences segmentation and clustering are usually set to be 3 – 5.

Secondly, the unsupervised Bayesian on-line learning algorithm is developed for the probability based multiple visual features fusion based image segmentation. In the Bayesian approach different visual features are combined with an adaptively weighting algorithm. Each visual feature cluster's PDF is assigned to be the Gaussian Distribution. Next the unsupervised Bayesian on-line learning algorithm is implemented for the image

features clusters' PDF parameters estimation. The learning goes on with every image by using the previous image segmentation information to compute the desired density $P(x|D)$.

Through the development procedures of the unsupervised Bayesian on-line learning approach, the Bayesian estimation method has more advantages than the Maximum-Likelihood estimation approach [46]. The Bayesian approach reflects the remaining uncertainty in the possibility models. The maximum-likelihood solution $P(x|\hat{\theta})$ is of the assumed parametric form, which is not for the Bayesian solution. In general, through the use of full $P(\theta|D)$ distribution the Bayesian methods use more of the information brought to the problem (such as more information about the $P(\theta|D)$ and the prior $P(\theta)$) than do the maximum-likelihood methods. If such information is reliable, the Bayesian methods can be expected to give better results. Furthermore, the Bayesian methods make more explicit the crucial problem of the bias and variance tradeoffs—roughly speaking, the balance between the accuracy of the estimation and its variance, which depend upon the amount of training data.

From the experimental results the unsupervised Bayesian on-line learning based algorithm is more accurate than other methods to segment the moving target from the image sequence with a changing background. The object in the clustered image is within the unique color and its shape is very accurate. The image segmentation algorithm can also well deal with the situations such as the target stopping, leaving the image scene or the object moving in the very cluttered environment and also the object's occlusion problem.

However there are also some obvious dangers associated with the decision-directed approach. If the initial image clustering is not reasonable good or if an unfortunate sequence of image pixels is encountered, the errors in clustering the unlabelled image pixels can drive the classifier into the wrong way, resulting in a solution corresponding roughly to one of the less peaks of the likelihood function. Even if the initial clustering is optimal, in general the resulting labelling will not be the same as the true class membership; the act of clustering will exclude the image pixels from the tails of the desired distribution and will include image pixels from the tails of the other distributions.

Thus if there are significant overlaps between the component densities, one can expect biased estimates and less-than-optimal results. Despite these drawbacks, the simplicity of decision-directed procedures makes the Bayesian approach a good choice for the parameters estimation. If the conditions are favorable, performances that is nearly optimal can be achieved at far less computational expense.

Considering the above dangers, it is found from the experimental testing that most of the unsupervised Bayesian on-line learning based image segmentation procedures work well if the following requirements are met. The parametric assumptions are valid (reasonable Gaussian Distribution Functions for the visual features clusters), and there is little overlap between the component densities (the clusters number can't be too small.), and the initial classifier design is roughly correct (the EM algorithm converges well.).

The learning process in the proposed algorithm uses the previous classification information to update the current parameters. Then it is possible for any errors to accumulate and hence to lead to situations that can be recovered from. This is one of the limitations of the proposed system. In fact this is also one of the limitations of the normal Bayesian estimation methods. The learning process uses the current clustered image's information to update the classification function. So if the current image's classification is correct, then the accumulated errors will be reduced and if the current image's segmentation is not accurate, then it will increase the errors. In fact this has been already considered as the future work, how to robustly set some criterions to update the classification functions with only the correct and useful information.

The video sequences tested in this Chapter are not quite long. In about 30 – 50 frames situations, from the experimental testing the stability of the segmentation algorithm is satisfactory. However the long-term stability should be studied and this is also considered as one of the future work.

In conclusion the unsupervised Bayesian on-line learning algorithm is simple to implement, robust to initial parameters and has a linear complexity. The linear complexity and good results from experiments make it a good choice for the object-based image segmentation system. Since the algorithm can make the accurate object based video images sequence segmentation when the cameras are also moving, it is suitable for the

proposed visual target tracking scheme in this thesis [43].

Chapter 6

Visual Features Localization: Template Updating and Matching

After the image visual features segmentation and clustering, the target's $2D$ positions in the images should be localized from the classified images, which is the third step of the proposed visual target tracking scheme. In this Chapter the research work on template matching is described. Two methods are presented here. One is the conventional Sum of Squared Difference (SSD) (correlation) based template matching method and the other is the novel template matching and updating approach based on the Bayesian probability fusion.

Template matching is a filtering method of detecting a particular feature in an image. Provided that the appearance of this feature in the image is known accurately, one can try to detect it with an operator called a template. This template is, in effect, a sub-image that looks just like the image of the object of interest. A similarity measure is computed which reflects how well the image data matches the template for each possible template location. The point of maximal match can be selected as the location of the feature.

Template matching techniques have many applications in image processing, pattern recognition, and video compression. For example, it is used for comparing different views of an object in pattern recognition, and for the motion estimation in video com-

pression algorithms. Strengths of template matching include its ease of implementation and efficient calculation over the entire set of locations. However, template matching is sensitive to changes in the object's shape, size, orientation, and change in the image intensities [53] [59]. In the proposed visual target tracking scheme, since both the cameras and the objects are moving, there are different motion regions in the classified images. In order to accurately localize the target's features in the 2D images, template matching methods are used here.

6.1 Normal Template Matching Method

In this thesis the template can just be obtained from the target's region in the clustered optical flow vectors field. In such a way the template consists of only the optical flow vectors of the moving target. The template does not contain any other special visual character or any model of the target. Template matching is normally used in the situations when there are several objects moving in the image scene. If assuming that the sizes of these objects are almost the same, the simple template matching methods can be used to find the required target.

6.1.1 SSD (Correlation) based Template Matching

One standard similarity measure between an image function $f(X)$ and a template $T(X)$ is the Sum of Squared Difference (SSD) using the Euclidean distance $d(Y)$, which is given by [222]

$$d(Y) = \sum_X [f(Y - X) - T(X)]^2 \quad (6.1)$$

where $\sum_X = \sum_{x=0}^M \sum_{y=0}^N$ and M, N are the sizes of the template.

If the image at the point Y is an exact match to the template, then $d(Y) = 0$. Otherwise, $d(Y) > 0$.

Expanding Eq.(6.1), it shows

$$d(Y) = \sum_X [f^2(Y - X) - 2f(Y - X)T(X) + T^2(X)] \quad (6.2)$$

where $\sum_X T^2(X)$ is a constant term. When $\sum_X f^2(Y - X)$ is approximately constant, these two terms can be neglected. Then the left part of Eq.6.2 is called the cross correlation between the f and T , shown as below

$$R_{Tf} = \sum_X f(Y - X)T(X) \quad (6.3)$$

which is maximized when the portion of the image “under” T is identical to f .

Next the template matching calculation can be visualized as imagining the template being shifted across the image to different offsets, then the superimposed values at this offset are multiplied together, and the products are added. This result of the sum of products forms an entry in the “correlation array” whose coordinates are the offsets attained by the template.

If the template is allowed to take all the offsets with respect to the image so that some overlap may take place, and the correlation array is larger than either the template or the image. An $n \times n$ image with a $m \times m$ template yields a $(n + m - 1) \times (n + m - 1)$ correlation array. If the template is not allowed to shift off the image, the correlation array is $(n - m + 1) \times (n - m + 1)$, here $m < n$. Another form of correlation results from computing the offsets modular the size of the image, i.e., the template “wraps around” the image. Being shifted off to the right, its right part will reappear on the left of the image. This kind of correlation is called the periodic correlation; otherwise, those without “wrap around” properties are called the aperiodic correlation.

6.1.1.1 Advantages and Shortcomings of the Correlation based Template Matching

Fig.(6.1) provides an example of the aperiodic “shift, add, multiply” template matching. Here the correlation is only calculated for offsets that leave the template entirely within

Template	Image	Correlation
1 1 1	1 1 0 0 0	7 4 2 x x
1 1 1	1 1 1 0 0	5 3 2 x x
1 1 1	1 0 1 0 0	2 1 9 x x
	0 0 0 0 0	x x x x x
	0 0 0 0 8	x x x x x
		x = undefined

Figure 6.1: Correlating an Image With Noise from [226]

the image. The correct peak is the upper left one at (0,0) offset. The “false alarm” at offset (2,2) is caused by the bright “noise point” in the lower right of the image. This figure illustrates some difficulties with this simple correlation measure of similarity. Many advantages and shortcomings of this measure result from the fact that it is linear. The advantages have to do with the existence of algorithm for performing the calculation efficiently (in a transform domain) for the entire set of offsets. The disadvantages have to do with the fact that the metric is sensitive to properties of the image that may vary with the offset, such as its average brightness, slight changes in the shape of the object, its size, orientation, or intensity values can also disturb the match [226].

The correlation based template matching is very important, especially if Eq.(6.3) is viewed as a filtering operation instead of an algorithm that does all the work of object detection. Choosing one or more templates (filters) that transform the image, certain features of an object are more apparent. These templates generally highlight the sub-parts of the objects. One example is the edge template.

In this thesis the correlation based template matching can be used to localize the position of the object of interest in the 2D images. The coordinates of the ROI are considered as the 2D position of the target in the image, which are used in the next Chapter for estimating the target’s 3D dynamics in the world coordinate. In the SSD matching the

template consists of only optical flow vectors, thus the extraction of the ROI is carried out only with the target's dynamic features, which can make the tracking system robust to track targets independent of their other visual features such as shape or color. For example, if a target is moving in the images and its color changes, template matching using color information will go wrong. In such a situation, the target's motion is still the same so template matching by optical flow vectors can give better results. Also using the 2D motion vectors instead of the image's 3D *RGB* color information reduces the whole system's computational cost.

At present, the initial template can be automatically identified from the clustered optical flow motion field if there is only one target moving or only one new object entering the image scene for tracking. If there are several objects moving together in the image, then the initial template must be selected manually from the first clustered optical flow vectors field. Since the matching template is initialized, the matching value should be very small when the target is moving in the image scene. If the matching value is too large than a preset threshold, it means that the target has exited and the tracker can then be automatically terminated. The tracker's automatic initialization, termination or multiple people tracking have been discussed with experimental results in Chapter 8. During the SSD matching process, the template is updated in real-time with the target's previous estimated 3D dynamic information. This updating process is known as the "naive update" template updating algorithm [133].

6.2 Bayesian Probability Fusion based Template Matching and Updating

The above part discusses the normal correlation based template matching. There are several problems with the simple correlation based template matching. The experimental performances from the correlation based template matching are normally not robust [91]. So in this Chapter a more accurate and robust template matching and updating algorithm, the Bayesian probability fusion based template matching, is proposed.

6.2.1 Introduction

Normally the object tracking is also the process of updating the object visual features over time. The complete set of the object's visual features includes position, motion, shape, appearance and so on. The appearance comprises of a set of visual features representing the object's region in a frame. To suppress noise and achieve the tracking stability, the visual features should be learned and updated during the tracking process.

One problem remains in the general template matching based target tracking. The underlying assumption behind template matching is that the appearance of the object remains the same throughout the entire image sequence. This assumption is generally reasonable for a certain period of time, but eventually the template is no longer an accurate model of the appearance of the object. If without a priori object appearance model (like the robust appearance model from [85]), the template has to be updated during matching. A "naive" solution to this updating problem is to update the template every frame (or every t frames) with a new template extracted from the current image at the current location of the template [133] [187]. The drawback existing in this "naive" template updating algorithm is that the template may drift away. Or if there exists any large shape rotation or appearance change of the tracked target, the template matching will fail. The "naive updating" will also fail at the presence of occlusions or abrupt environmental changes such as lighting conditions [133].

To make the tracking robust to the above factors, a more persistent template updating and matching scheme should be developed. The method needs to be capable of memorizing the template's appearance features and adjusting the matching functions through the image sequence. In such a way the template matching can be insensitive to sudden changes of the object appearance and, at the same time, be able to adapt to slow appearance changes. The problems in template matching such as template drifting, shape rotation, appearance change or occluded matching can then be solved.

Most recently Ross et. al. [120] [121] [176] have addressed the above target tracking problems. They have also concluded that most visual tracking algorithms are operating on the premise that the models of the objects being tracked are invariant to internal appearance changes or external variations such as lighting or viewpoint. Consequently

these tracking algorithms can not update the models once they are built or learned at the outset. Instead of using a simple contour to enclose an image region or color pixels to represent the moving object for target tracking, they use an eigenbasis to represent the object being tracked. Their methods incrementally learn and adapt a low dimensional eigenspace representation to reflect the appearance changes of the target, thereby facilitating the tracking tasks. Their methods well tackle the problems in handling the appearance variability of a target. However there are some drawbacks. First, the target will be drifted off by some pixels before recovering in the later frames. Secondly, their methods can't handle occlusion well.

In [235], Zhou et. al. present an approach that incorporates appearance-adaptive models in a particle filter to realize robust visual tracking and recognition algorithms. To stabilize the tracker, they propose the following modifications: an observation model arising from an adaptive appearance model, an adaptive velocity motion model with adaptive noise variances, and an adaptive number of particles in the particle filter. However in their method all the models are adaptive ones. Adaptiveness means a lot of tuning parameters and less robustness due to the complexity. Also as far as the computational load is concerned, the stochastic algorithms in general are more intense. The stochastic approaches often lead to optimization problems too. Appearance-adaptive modes and particle filters are time-consuming stochastic methods, which make their whole system run very slowly.

In this Chapter a fusion based template matching and updating algorithm is developed to solve the above problems [91]. The idea of combining different methods for target tracking has been proposed in [131]. From [115] it is clear that multiple information can be easily and efficiently combined according to the rules of the Bayesian probability theory [129] by using Probability Distribution Functions (PDF). Through Bayesian probability based fusion, different methods can combine the advantages of each one and compensate their drawbacks with a weighting algorithm. Thus in this Chapter the Bayesian probability theories are used to combine different template matching methods.

Concluded from the previous work on Bayesian based template matching [46] [95] [115] [209], the Bayesian probability theory is flexible to incorporate different template match-

ing methods, if the probability function of each method is correctly chosen. Here the different template matching methods can be obtained by matching individual features such as the object's size, image texture, colors, etc. Also Bayesian theory provides the way to fuse two independent probabilities into one equation to give the joint probability for the Maximum Likelihood estimation [46], which is rather simple in implementation and faster in calculation and also allows the fusion method to result in an efficient and better template update in the presence of rapid illumination and shape changes.

In this Chapter the method based on the Bayesian probability theory is proposed to fuse two commonly used matching methods for a robust template matching. Two different template matching methods (Sum of Squared Errors and Gaussian Mixture Models) are combined through the Bayesian probability theory. Each method is assigned with a Gaussian PDF to denote the probability of matching in the image. The joint PDF is then used for the Maximum Likelihood algorithm to find the maximum matching probability, which gives the template's best matched region in the images.

The Sum of the Squared Errors (SSE) template matching is to locate the template in the image with the minimum Sum of the Squared Difference (SSD) between the image and the template's pixel intensity value. SSE method is normally considered as the basic and accurate template matching method, which is also simple to implement. In this tracking system in order to get the Bayesian based fusion, the probability of this template matching is assigned with a distribution function. Here the idea is that the ratio between the template's SSE matching value in the template's each possible location in the image and the sum of the template's pixels intensity value obeys an Gaussian Distribution Function. (The details of this distribution will be discussed later in the chapter.) The maximum value of this PDF means a best template matching in the image by SSE.

If the SSE method is used, another method incorporating with it should be able to memorize the template appearances and should also be easy to find the template matched region through the Bayesian probability theory. Here the second template matching method is selected based on the template appearance modelling or learning. Now some work have been done on the object tracking methods by smoothing or learning the

object's appearance features. The appearance feature types range from the gray value of pixels [205], phase data from wavelet filters [85], to global statistics like mean color [219] or histogram [211]. From the most recent work of [151], the target's appearance template can be smoothed temporally by a robust Kalman Filter during tracking. With the robust Kalman Filtering, outliers due to the partial occlusions can be downweighted by an observation model using a robust error norm like "the Mahalanobis distance". However these kinds of smoothing or filtering methods are assumed with a tracking filter to update and smooth the template pixel by pixel. They require a lot of computational time and are not easy to implement. And the robust Kalman Filter does not accept the long time partial occlusion problems, because the template filter will slowly accept the data at the occluded area.

Instead of smoothing the template, in the proposed template matching scheme the Gaussian Mixture Models (GMMs) is used as the proposed template representation scheme. Gaussian Mixture Models (GMMs) is among the most statistically mature methods for clustering (though they are also used intensively for the density estimation) [63] [163]. GMMs can also be employed to model the visual features of an object in order to perform tasks such as the real-time target tracking [168] [195]. GMMs also provides the mixture Gaussian distribution functions for the matching probability estimation, which can be used later for the Bayesian fusion part.

In this Chapter GMMs is used to represent the template's appearance as a mixture of different Gaussian models (the 3 color components [169]). Since the template is modelled by GMMs, its appearance changes can be measured by the PDF parameters. These parameters (usually known as the mean and variance) can be used to remember and constraint the template to its original or past appearances when updating the template. Thus GMMs has the ability to remember the target's visual features whenever the orientation, zoom factor or brightness of the tracked object changes. In this way it can solve the template's shape rotation, appearance changes and occlusion matching problems. In this Chapter the EM algorithm [63] [163] is implemented to initialize the template's GMMs parameters in the first image frame.

In order to make the accurate template matching, the template needs to be updated

between two consecutive image frames. In this Chapter the updating uses not only the previous template but also its previous matching probability parameters. The updating can recognize the changes of the template throughout the image sequence and make the template matching more robust. In this Chapter the object's template is represented by the Gaussian Mixture Models. During the matching, the template GMMs parameters are updated frame by frame, which learn the template's appearance changes and make robust matching. The updating algorithm here improves the work of [133] by solving the problems of the template shape rotation, appearance change and occluded template matching.

6.2.2 Bayesian Probability based Two Template Updating and Matching Algorithms Fusion

6.2.2.1 Bayesian Probability based Template Matching Algorithms Fusion

This part proposes the algorithm to fuse two different methods for template matching in the images. First let the template's location in I_t (t denotes the time index) be denoted by L_t , $1 \leq L_t(x, y) \leq k$. k is the total number of the template's possible locations in the image I_t . Let the probability of the template T_t in the frame I_t belonging to one of the possible locations $l_{t,i}$ ($1 \leq i \leq k$) be given by $P(l_{t,i}|T_t)$. Here if the template is T_t , when each region $l_{t,i}$ is matched to this template, $P(l_{t,i}|T_t)$ is the matching probability value. Now from the Bayesian theory,

$$P(l_{t,i}|T_t) = \frac{P(T_t|l_{t,i})P(l_{t,i})}{P(T_t)} \quad (6.4)$$

This equation relates the posteriori probability $P(l_{t,i}|T_t)$ to the product of $P(T_t|l_{t,i})$ and $P(l_{t,i})$ with the scale factor $P(T_t)$. This equation is used to find the probability of the template belonging to each of the possible region $l_{t,i}$ and then assign the template to the region that returns the maximum probability. So the template is assigned with the

maximum probability as shown in:

$$L_t = \arg \max_i \left\{ \frac{P(T_t|l_{t,i})P(l_{t,i})}{P(T_t)} \right\} \quad (6.5)$$

where $1 \leq i \leq k$. Here $P(T_t|l_{t,i})$ is called the likelihood of $l_{t,i}$ with respect to T_t , a term chosen to indicate that, other things being equal, the possible $l_{t,i}$ for which $P(T_t|l_{t,i})$ is large is more “likely” to be the best matched region. Next the denominator in Eq.6.5 can be ignored because it does not depend on i and is always positive. Then with the log function, the likelihood relationship is:

$$L_t = \arg \max_i \{ \ln(P(T_t|l_{t,i})) + \ln(P(l_{t,i})) \} \quad (6.6)$$

Here the effects of the prior term $\ln(P(l_{t,i}))$ in Eq.(6.6) may be ignored if all the template’s possible locations are equally likely to be observed.

Since the two matching algorithms used in this Chapter (Sum of Squared Errors (SSE) template matching and Gaussian Mixture Models (GMMs) template matching) are done in the individual and separate matching processes, their matching probabilities are assumed to be independent. Individual weights ω are used for each template matching component. Then the log-likelihood equation can be written as (Eq.(6.7))

$$L_t = \arg \max_i \{ \omega_t^{SSE} \ln P(T_t|l_{t,i}, SSE) + \omega_t^{GMMs} \ln P(T_t|l_{t,i}, GMMs) \} \quad (6.7)$$

where ω_{SSE} , ω_{GMMs} denote the weights for the matching components of SSE, and GMMs. $P(T_t|l_{t,i}, SSE)$ denotes the matching PDF from the SSE template matching. $P(T_t|l_{t,i}, GMMs)$ is the matching PDF from the GMMs matching algorithm. Individual weights ω are used for each set of matching PDFs (Eq.6.7).

Here the weights are estimated from the experimental test and trial, which are mainly based on the understanding about the desired template matching performance. If the matching by SSE is ‘reliable’, then the SSE term is weighted more, and correspondingly decrease the weight of the GMMs term, and vice versa. Thus the weights can be automatically selected based on the computation of $\ln P(T_t|l_{t,i}, SSE)$ and $\ln P(T_t|l_{t,i}, GMMs)$.

The higher the probability $\ln P(T_t|l_{t,i}, SSE)$ is, the better the template matches to the images by SSE. If the probability $\ln P(T_t|l_{t,i}, SSE)$ is low, it indicates that a higher weight should be given to the GMMs for template matching. For examples, when the object's visual features such as RGB colors do not change greatly, GMMs can have large weights. If the environmental condition is changing, the lighting change makes the object look different in different frames, then the SSE needs to be weighted more. The weights here can be estimated using the same procedures from the previous Chapter 5.3.2.3, because the fusion strategies in these two methods are some kind of the same.

In order to get the maximum value of Eq.(6.7), each matching algorithm's probability function $P(T_t|l_{t,i}, SSE)$ and $P(T_t|l_{t,i}, GMMs)$ should be known with every frame. Here the first frame from the video file is matched by using the matching scheme which is called the initial template matching. Then the matching for the whole image sequence is computed iteratively by using the updated template information from the previous images.

6.2.2.2 Sum of Squared Errors Template Matching

The Sum of the Squared Errors is a normally used method to measure the degree of the matching between the template and the image frame.

First let ϕ_{t-1} be the possible template's transformation matrix (also can be called "warp") from the template region Ω to the object region and p_{t-1} be the parameter vector of this transformation at time $t - 1$. $x' = \phi(x; p_{t-1})$ denotes the transformation of the point $x \in \Omega$. Generally the template's motion of skew, aspect ratio change, translation, rotation and scaling are considered, so the vector p has six parameters $\{p_1, p_2, p_3, p_4, p_5, p_6\}$. The transformation equation can be written as:

$$\begin{aligned} \phi(x; p) &= \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = \begin{pmatrix} (1 + p_1) \cdot x_1 + p_3 \cdot x_2 + p_5 \\ p_2 \cdot x_1 + (1 + p_4) \cdot x_2 + p_6 \end{pmatrix} \\ &= \begin{pmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \end{aligned} \quad (6.8)$$

The transformation ϕ decides the template's possible shift or shape transformation between two consecutive images. This transformation ϕ is included in the next section to develop the template updating algorithm.

To reduce the system complexity, the motion types of the object's shape transformation in the image plane is restricted by a global transformation known as the affine transformation (Eq.6.8). This assumption is justified for a large range of tracking applications where the object motion is rigid or the object is sufficiently distant from the camera [151]. By doing so, a separate model for the object shape is not required since this can be determined from the object motion. Here the 2D affine warp is only valid when only 2D translations or in-plane 3D rotations of planar objects (which is located far from the camera) are considered, but not valid for the general 3D rigid motion of general 3D objects.

Next the matching can be performed by finding the image region that yields the sum of squared errors with respect to the template of T_t [133]. This is achieved with the minimization:

$$\hat{p}_t = \arg \min_p \sum_{x \in \Omega} [I_t(\phi(x; p)) - T_t]^2 \quad (6.9)$$

where $I_t(\phi(x; p))$ is the feature vector observed in the current image at the point $\phi(x; p)$. $I_t(\phi(x; p))$ can be considered as the template possible location in the image. The summation $\sum_{x \in \Omega}$ is over all the pixels in the template region. Target matching using robust error functions has been used and has been more heavily studied in [3]. The minimization can be efficiently performed by the gradient descent algorithm in a coarse-to-fine manner.

In this thesis the SSE matching is assigned with a probability distribution function to find the best template matched region in the image. This PDF is used later as $P(T_t | l_{t,i}, SSE)$ for the Bayesian probability fusion in Eq.6.7. Here the idea is that it is assumed that the ratio of the template matching SSE value $\sum_{x \in \Omega} [I_t(\phi(x; p)) - T_t]^2$ from Eq.6.9 and the sum of pixels intensity value of the template $\sum T_t$ obeys a Gaussian

Distribution, which means

$$\begin{aligned}
 p(T_t|SSE) &= N\left(\frac{\sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2}{\sum T_t}, \mu_{SSE,t}, \sigma_{SSE,t}\right) \\
 &= \frac{1}{\sqrt{(2\pi)\sigma_{SSE,t}^2}} e^{-\frac{1}{2\sigma_{SSE,t}^2} \left(\frac{\sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2}{\sum T_t} - \mu_{SSE,t} \right)^2} \quad (6.10)
 \end{aligned}$$

where $\mu_{SSE,t}$ and $\sigma_{SSE,t}$ are the PDF mean and variance parameters. In this Chapter these parameters are first initialized through the initial template matching. Then they are updated frame by frame using the template updating algorithm developed in this Chapter. The maximum value of Eq.6.10 means the best template matching by SSE.

The use of SSE in this thesis is slightly different from the standard SSD approach mainly treating the SSE divided by the sum of the template image pixels as a Gaussian probability distribution function (PDF). This different way using SSE is quite necessary for the Bayesian probability based fusion. The Gaussian assumption in Eq.6.10 is justified with the following reasons. First, for the template matched to the image I_t , its sum of the pixel's intensity value $\sum T_t$ should be proportioned to the first template's $\sum T_1$. This is because the templates in different image frames always have roughly the same visual appearance. Secondly, since the template is always matched to the region with the mostly same appearance as itself, the sum of the matched region's pixels intensity value is close to the sum of the template's pixels intensity value. Their difference has the smallest SAD value and is also roughly proportional according to the template's changes in images. Thus when the template is matched to the image, the minimum value of SSE $\sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2$ (best matched area) should also be proportioned to the previous template best matched region's SSE value. The above two proportions have the same proportion parameter (according to the template's changes). Now based on the above explanations it can be just written as:

$$\begin{aligned}
 \sum T_t &= \kappa \sum T_1 + \alpha \\
 \sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2 &= \kappa \sum_{x \in \Omega} [I_1(\phi(x;p)) - T_1]^2 + \beta \quad (6.11)
 \end{aligned}$$

Here κ is the proportion parameter between the sum of the pixels intensity value of the template T_t and the first template T_1 . The same κ is also the proportion parameter between the two matching SSE values. As the proportion is only the rough approximation, α and β are added here as the noise parameters (Eq.6.11). Now divide the two equations from Eq.(6.11) in both sides, it gives:

$$\begin{aligned} \frac{\sum_{x \in \Omega} [I_t(\phi(x; p)) - T_t]^2}{\sum T_t} &= \frac{\kappa \sum_{x \in \Omega} [I_1(\phi(x; p)) - T_1]^2 + \beta}{\kappa \sum T_1 + \alpha} \\ &= \frac{\kappa \sum_{x \in \Omega} [I_1(\phi(x; p)) - T_1]^2}{\kappa \sum T_1} \cdot \frac{1 + \frac{\beta}{\kappa \sum_{x \in \Omega} [I_1(\phi(x; p)) - T_1]^2}}{1 + \frac{\alpha}{\kappa \sum T_1}} \\ &\approx \frac{\sum_{x \in \Omega} [I_1(\phi(x; p)) - T_1]^2}{\sum T_1} \end{aligned} \quad (6.12)$$

$$\frac{\alpha}{\kappa \sum T_1} \approx 0 \quad \frac{\beta}{\kappa \sum_{x \in \Omega} [I_1(\phi(x; p)) - T_1]^2} \approx 0 \quad (6.13)$$

Here since α and β are much smaller values considered as the noise effect, their value can be approximated to be zero and then the ratio values in Eq.6.13 are zeros. Thus from Eq.6.12 the matching probability parameter $\frac{\sum_{x \in \Omega} [I_t(\phi(x; p)) - T_t]^2}{\sum T_t}$ should have roughly the same value through the whole image sequence (approximately equal to the first template's $\frac{\sum_{x \in \Omega} [I_1(\phi(x; p)) - T_1]^2}{\sum T_1}$ value) whenever the template is updated or the image frame is changing. Other possible template region in the image will give quite different $\sum_{x \in \Omega} [I_t(\phi(x; p)) - T_t]^2$ values, which thus can not give the ratio value close to the previous matched region's ratio value. So if $\frac{\sum_{x \in \Omega} [I_t(\phi(x; p)) - T_t]^2}{\sum T_t}$ is assumed with a Gaussian Distribution (The mean and variance can be obtained from the initial template matching process.), the region with the maximum value of this PDF (Eq.6.10) means a best template matching in the image by the SSE algorithm.

Also from the experimental testing, the use of SSE in a Gaussian distribution framework can provide the similar template matching results as the standard SSD approach. Thus based on both theoretical and experimental justifications, in this Chapter the SSE is treated in a PDF way for the Bayesian fusion based template matching and updating.

6.2.2.3 Gaussian Mixture Models for Template Matching

At this stage, the image pixels are grouped into homogeneous regions known as the template. Here it is assumed that the template's color distributions are generated by a mixture of Gaussians. Then the object's template appearance model is modelled as the Gaussian mixture distributions in the 3D RGB feature space. The GMMs distribution of a random point x in the image template T_{GMMs} is a mixture of 3 color Gaussians if its density function is:

$$T_{GMMs}(x|\theta) = \sum_{j=1}^3 \alpha_j \frac{1}{\sqrt{(2\pi)^d} |\sigma_j|} e^{-\frac{1}{2}(x_j - \mu_j)^T \sigma_j^{-2} (x_j - \mu_j)} \quad (6.14)$$

where α_j is the mixing proportion, the parameter set $\theta = \{\alpha_j, \mu_j, \sigma_j\}_{j=1}^3$ consists of

1. $\alpha_j > 0, \sum_{j=1}^3 \alpha_j = 1$
2. $\mu_j \in R^d$ and σ_j is a $d \times d$ positive definite matrix (here $d = 1$ for the case of the pixel's each color component Gaussian Distribution Function.).

Now with GMMs the template can be represented as a combination of three color ellipses, each of which centers at the mean μ_j . The variance matrices σ_j determine the geometric characteristics of the ellipses [63]. Since the template's color features are modelled by the GMMs, no matter how the target's visual features changes, the GMMs has the ability to remember the visual features by using the GMMs mean and variance parameters. In such a way, GMMs can help the other matching method (SSE) to solve the template matching problems such as the template's drifts, shape rotations, appearance changes or occlusions.

The template matching by GMMs can be obtained through the combination of matching by each color's Gaussian Probability Distribution function. In this Chapter since the three colors are separately extracted from the images as the R , G , B colors and they can be independently matched, the GMMs matching probability can be defined as the joint product of each different Gaussian Distribution Functions. Image colors can be normally represented in several different color spaces such as RGB or YUV . Here the RGB color space is selected. The RGB colors of each image pixel are unpredictable and

can be randomly mixed so it is reasonable to assume that the RGB colors distributions are independent for each color channel. Here the mean μ_j and variance σ_j 's value from Eq.6.14 can be used to estimate the matching accuracy with the Bayesian probability theory for $P(T_{GMMs}|GMMs)$.

$$P(T_{GMMs}|GMMs) = P(T_{GMMs,1}|GMMs_1) * P(T_{GMMs,2}|GMMs_2) * P(T_{GMMs,3}|GMMs_3) \quad (6.15)$$

$$\begin{aligned} P(T_{GMMs,i}|GMMs_i) &= N(\mu_{I_t(\phi(x;p)),i}, \mu_{GMMs,i}, \sigma_{GMMs,i}) \\ &= \frac{1}{\sqrt{(2\pi)\sigma_{GMMs,i}^2}} e^{-\frac{1}{2\sigma_{GMMs,i}^2}(\mu_{I_t(\phi(x;p)),i} - \mu_{GMMs,i})^2} \end{aligned} \quad (6.16)$$

where $1 \leq i \leq 3$. $T_{GMMs,i}$ denotes different component of the color GMMs. In the above equation, $\mu_{I_t(\phi(x;p))}$ is one of the template's colors' PDF means from the template's possible region ($I_t(\phi(x;p))$) in the image. The region with the maximum value from Eq.6.16 means the best template matching result by GMMs.

The GMMs' PDF parameters in Eq.6.16 are initialized iteratively with the EM algorithm [39] during the template initialization process. These parameters are then used for the next image template matching.

6.2.2.4 PDF Parameters Initialization for GMMs

The initial template is selected manually from the first image, and at the same time the GMMs' PDF parameters are initialized iteratively with the EM algorithm [39]. These parameters are then used for the next image template matching. Now given a set of feature vectors $x_1, \dots, x_n$, the maximum likelihood estimation of θ is:

$$\theta_{ML} = \arg \max_{\theta} L(\theta|x_1, \dots, x_n) = \arg \max_{\theta} \sum_{i=1}^n \log T_{GMMs}(x_i|\theta) \quad (6.17)$$

Here the parameter θ consists of the mean μ_j and covariance σ_j of the GMMs. The Expectation-Maximization (EM) algorithm is used to estimate these maximum likeli-

hood parameters for the first template GMMs' PDF parameters initialization.

Given the current estimation of the parameter set θ , each iteration of the EM algorithm re-estimates the GMMs parameter set according to the following two steps:

1. Expectation Step:

$$\omega_{ij} = \frac{\alpha_j \sum_{x_n \in \Omega_i} T_{GMMs}(x_n | \mu_j, \sigma_j)}{\sum_{l=1}^3 \alpha_l \sum_{x_n \in \Omega_i} T_{GMMs}(x_n | \mu_l, \sigma_l)} \quad (6.18)$$

Here $j = 1, \dots, 3$, $i = 1, \dots, k$, $n = 1, \dots, m$, and i indicates the template's possible locations in the image. x_n is the n th point in the possible template region Ω_i in the image. Here the template is assumed to have m points in the image. The term ω_{ij} is the posterior probability that the feature vector x_i is sampled from the j th component of the mixture distribution.

2. Maximization Step:

$$\begin{aligned} \hat{\alpha}_j &\leftarrow \frac{1}{3} \sum_{i=1}^3 \omega_{ij} \\ \hat{\mu}_j &\leftarrow \frac{\sum_{i=1}^3 \omega_{ij} \frac{\sum_{x_n \in \Omega_i} x_n}{m}}{\sum_{i=1}^3 \omega_{ij}} \\ \hat{\sigma}_j &\leftarrow \frac{\sum_{i=1}^3 \omega_{ij} \left(\frac{\sum_{x_n \in \Omega_i} x_n}{m} - \hat{\mu}_j \right) \left(\frac{\sum_{x_n \in \Omega_i} x_n}{m} - \hat{\mu}_j \right)^T}{\sum_{i=1}^3 \omega_{ij}} \end{aligned} \quad (6.19)$$

In this study, the convergence is based on the log-likelihood measure, and a 1 percent threshold is selected. Other possible convergence options include using a fixed number of iterations of the EM algorithm or defining the target measures. It is found experimentally that the above convergence methodology works well for the experimental purpose.

6.2.3 Minimization of the Bayesian Fusion Equation

Now after the two different template matching PDFs are obtained (Eq.6.10 and Eq.6.16), the maximization of the Bayesian fusion equation from Eq.6.7 can be calculated as:

$$\begin{aligned}
L_t &= \arg \max_i \{ \omega_t^{SSE} \ln P(T_t | l_{t,i}, SSE) + \omega_t^{GMMs} \ln P(T_t | l_{t,i}, GMMs) \} \\
&= \arg \max_i \left\{ \omega_t^{SSE} \left(-\frac{(\frac{\sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2}{\sum T_t} - \mu_{SSE,t})^2}{\sigma_{SSE,t}^2} \right) \right. \\
&\quad + \omega_t^{GMMs} \left(-\frac{(\mu_{I_t(\phi(x;p)),1} - \mu_{GMMs,1})^2}{\sigma_{GMMs,1}^2} \right) \\
&\quad + \omega_t^{GMMs} \left(-\frac{(\mu_{I_t(\phi(x;p)),2} - \mu_{GMMs,2})^2}{\sigma_{GMMs,2}^2} \right) \\
&\quad \left. + \omega_t^{GMMs} \left(-\frac{(\mu_{I_t(\phi(x;p)),3} - \mu_{GMMs,3})^2}{\sigma_{GMMs,3}^2} \right) \right\} \\
&= \arg \min_i \left\{ \omega_t^{SSE} \left(\frac{(\frac{\sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2}{\sum T_t} - \mu_{SSE,t})^2}{\sigma_{SSE,t}^2} \right) \right. \\
&\quad + \omega_t^{GMMs} \left(\frac{(\mu_{I_t(\phi(x;p)),1} - \mu_{GMMs,1})^2}{\sigma_{GMMs,1}^2} \right) \\
&\quad + \omega_t^{GMMs} \left(\frac{(\mu_{I_t(\phi(x;p)),2} - \mu_{GMMs,2})^2}{\sigma_{GMMs,2}^2} \right) \\
&\quad \left. + \omega_t^{GMMs} \left(\frac{(\mu_{I_t(\phi(x;p)),3} - \mu_{GMMs,3})^2}{\sigma_{GMMs,3}^2} \right) \right\} \tag{6.20}
\end{aligned}$$

The minimization of Eq.(6.20) is obtained from the template warp parameter p (Eq.(6.8)). Here the gradient descent method can be implemented to find the minimization value of Eq.(6.20) by

$$p \leftarrow p + \Delta p \tag{6.21}$$

The available gradient descent approaches for the convergence estimation of parameter p have been reviewed in [3]. All the methods available in [3] are equivalent in the sense that they take the same steps in each iteration to the first order in Δp . Here the inverse compositional approach can be selected because it is simpler and far more computationally efficient than the other methods according to the review paper.

In the experiments as the gradient descent algorithm finds a local minimum only, it may

result in a wrong location when the warp parameter p is too large. To overcome this disadvantages, the gradient descent method is repeated with a set of initial values of p with different components and the best resulting location is selected. Once p has been obtained, the target's location in the image can be defined as $I_t(\phi(x; p))$.

6.2.4 Template Updating Strategy

The template updating problems have been studied in [133]. In this Chapter the algorithm is extended to make more robust template updating with the Bayesian probability fusion theory.

First let the template be updated with the normal “naive update” method (the correlation based template matching also uses this updating strategy.), which sets the new template to be the region of the input image that the template is tracked in $I_{t-1}(x)$

$$\hat{T}_t(x) = I_{t-1}(\phi(x; p_{t-1})) \quad t \geq 1 \quad (6.22)$$

From [133] with the “naive updating” method, the template may eventually drift away from the original object. Small errors in the previous warp parameter p_{t-1} mean that the new template $I_{t-1}(\phi(x; p_{t-1}))$ is always a slighted transformed version of what it ideally should be. These errors accumulate and, after a while, the template drifts away from the object that it is initialized to track. Note that simple variants of this strategy such as updating the template every few frames, although more robust, also eventually suffer from the same drifting problem. In the method of [133] they retain the first template and use the first template aligned with the template updated from the “naive update” to solve the template drift problem and late give the final matching.

In this Chapter the template updating method is done with the Bayesian probability theory. The template updated from the “naive update” $\hat{T}_t(x)$ (Eq.6.22) with the SSE and GMMs matching PDF parameters from the image I_{t-1} ($\mu_{SSE,t-1}$, $\sigma_{SSE,t-1}$, $\mu_{GMMs,t-1,i}$ and $\sigma_{GMMs,t-1,i}$) is first matched to the current image I_t . The maximum matching probability is obtained through Eq.6.20. Next the original first image template T_1 with the same matching PDF parameters from the image I_{t-1} is matched

to image I_t , which gives another matching probability. Here these two probabilities $\arg \max_i P(\hat{T}_t(x)|l_{t,i}, SSE_{t-1}, GMM_{s_{t-1}})$ and $\arg \max_i P(T_1|l_{t,i}, SSE_{t-1}, GMM_{s_{t-1}})$ are compared, the higher value means a better matching. Thus the template with the higher matching probability value is selected as the template for the current image template matching. At the same time the SSE and GMMs matching PDF parameters with this template are estimated for the next image frame template updating and matching.

Through this updating, the template's SSE matching and GMMs matching are considered at the same time. Their joint matching probability gives a solution to the template drifting problems and also the template's shape rotation, appearance change or occlusion problems. Here the two matching PDF parameters are also updated frame by frame. The matching PDF parameters adjust the matching requirement with the template's appearance changes in the image sequence.

The template updating method proposed here needs a two-step template matching (first the matching is done with the template from the "naive" update and then the matching is done with the first image template) to decide a better template for updating. It requires more computational time. In order to reduce the computational time, a threshold can be set beforehand to decide whether the naive updated template can be kept or not. Here according to the experimental testing, the best matched region's matching probability value from the proposed method is almost at the same order of magnitude. The template's other possible locations will give much smaller matching probability values compared to the best matching value. Thus a threshold can be set close to this normal best matching value. If the template matching probability value is larger than this threshold, thus the matched region can be just kept and used for the next image frame template matching ("naive update"). If the matching result is smaller than this threshold, then the template should be updated with the Bayesian fusion based algorithm developed in this Chapter.

6.2.5 Initial Template Matching

The initial template is selected manually from the first image frame and it roughly indicate the target's region in the image. Then the template GMMs' PDF parameters

(the mean μ and variance σ) are estimated from the EM algorithm. The SSE matching PDF parameters are estimated iteratively as follows. First the initial SSE parameters μ_{SSE} is assumed zero and σ_{SSE} is set by the experimental trial and test (normally it is set to be 0.01.). Then the template is matched to the image with the Bayesian algorithm developed in this Chapter. After the first time matching, the best matched area gives the SSE matching probability parameter μ_{SSE} from the calculations of $\frac{\sum_{x \in \Omega} [I_t(\phi(x;p)) - T_t]^2}{\sum T_t}$. The estimated parameters and the matched region are thus used for template matching of the same image for a second time. The matching continues in the same way iteratively until μ_{SSE} and σ_{SSE} converge to some values. The converged parameters and the target's located region are used later for the second image frame template matching.

6.3 Experimental Results

After the image segmentation and clustering from the previous Chapter, the major motion regions in the whole optical flow vectors field are clustered into several clusters. So here first, the SSD based template matching is carried out to identify the target by calculating the Euclidean distance between the template's optical flow vectors region and the clustered optical flow vectors field.

Here the SSD matching is between the template's optical flow vectors region and the clustered optical flow vectors field. The template is not matched to the whole optical flow vectors field. The clustered field from the image segmentation and clustering results is used here because it contains the major motion regions and removes the small noise areas. In such a way, the SSD template matching is more accurate and faster.

A template from the segmented optical flow vectors field is shown as Fig.6.2.1. The template contains only the optical flow vectors from the ROI. The matching values of the template are shown graphically in Fig.6.2. Here for the Euclidean distance based matching, the template is updated frame by frame by using the previous estimated ROI. This "naive template update" [133] process is: using the estimated 3D position of the target at time k , the flow vectors region (i.e. ROI) in the whole optical flow vectors field at time k is then extracted by an pinhole camera projection model. This flow vectors

Figure 6.2: The template from the clustered optical flow vectors field and the Euclidean distance based matching result. The smallest value in Fig.6.2.2 indicates the template's best matched region in the images.

region is used as the template for the image template matching at time $k + 1$.

Next in order to evaluate the performance and adjust the parameters of the proposed method, the Bayesian fusion based template matching algorithm has been tested on different image sequences and also compared with other methods.

From Fig.6.3 a blue target is moving in the image sequence. Later it is occluded by another stationary object. From the experimental results, the template matching performs well and the object is correctly tracked.

In Fig.6.4 the blue object is occluded by another moving object. The template matching results are also good and the object is precisely tracked.

The matching results in Fig.6.5 are bad because the template drifts away from the original template due to the “naive” template updating. Fig.6.6 shows the results when the template is updated with the robust Bayesian fusion based template updating algorithm. The first template and the template updated from the “naive” updating algorithm are considered at the same time for template updating, which solves the template drifting

Figure 6.3: The template matching results when the moving object is occluded by another stationary object. The blue object in the first frame is considered as the initial template.

Figure 6.4: The template matching results when the object is occluded by another moving object. The blue object in the first frame is considered as the initial template.

problem.

Fig.6.7 shows good results of the proposed algorithm when the object has great rotational motion in the 2D images. Here the target (the black object) is rotating in two different directions. In such situations, the template matching is really difficult since even between two consecutive image frames the template's shape and appearance change a lot. If only using the normal methods like SSE template matching (the algorithm in [133]) or

Figure 6.5: The template matching results when the black object is moving towards the camera. Here the template is not updated with the Bayesian fusion algorithm developed in this Chapter. Thus finally the template drifts away. The matching results are not good. The black object in the first frame is considered as the initial template.

Figure 6.6: The template matching results when the object is moving towards the camera. The template matching performance is improved with the proposed robust template updating algorithm.

GMMs template matching, it is impossible to accurately locate the object because the object rotates a lot. The minimal value from SSE may only be the local minimal which is not the global minimal value for the object's correct region with the rotations. However in the proposed Bayesian fusion based algorithm, the GMMs is able to remember the template's visual features properties. Although its shape rotates, its other visual fea-

Figure 6.7: The template matching results when the object's shape has some changes. Here the black target is rotating greatly in two different directions. The black object in the first frame is considered as the initial template. The proposed template matching algorithm is to locate the template's most matched region in the images when the object has great rotations. The proposed Bayesian fusion based method gives good result.

tures properties (such as RGB color properties) remain the same throughout the image sequence. Thus combined SSE with GMMs, the Bayesian fusion based algorithm makes weighting between these two matchings and gets the best template matching results. As shown in Fig.6.7 the template's matched region in the image sequence is adjusted frame by frame. Finally the template matched region changes its own shape and good matching performances are achieved.

Figure 6.8: The template matching results when the object has some rotations. Compared with Fig.6.7, the template matching performance is bad if the template is not matched and updated by the algorithm developed in this Chapter.

Fig.6.8 shows the bad template matching results of the same image sequence of Fig.6.7. Fig.6.8.1, Fig.6.8.2, Fig.6.8.3 and Fig.6.8.4 are the results from the “naive” template updating. The template finally drifts away. Fig.6.8.5, Fig.6.8.6, Fig.6.8.7 and Fig.6.8.8 are the results when the GMMs component weights more and becomes the dominant factor of the matching. In such a way the matching results may converge as shown from Fig.6.8.5 to Fig.6.8.8. When the GMMs weights more, the algorithm tries to converge to the areas with less color changes. Thus it becomes a line as shown in Fig.6.8.8. With such a line, the GMMs matching probability values are always the same. Compared with these figures, the results from Fig.6.7 show good performances of the proposed Bayesian fusion method in this difficult matching situation.

Some other experimental results are presented in different situations in Fig.6.9, Fig.6.10 and Fig.6.11. Fig.6.9 deals with the situation when the object is leaving the image scene. Fig.6.10 gives the result when both the camera and object are moving. Fig.6.11 solves the occlusion tracking problem. The backgrounds in Fig.6.10 are cluttered and continuously changing.

In the multi-color persons matching results shown in Fig.6.10 and Fig.6.11, the GMMs models the templates’ complex color properties accurately and the template matching

Figure 6.9: The template matching results when the blue object leaves the image scene. The blue object with its full shape from the first frame is considered as the initial template. The template matching performance is good with the proposed method. The template matched region is changing due to the changes of the object's size in the image scene.

Figure 6.10: The person is moving with a moving camera. The person in the first image frame is considered as the initial template. The proposed Bayesian fusion based algorithm is used to locate the person in the image sequence. Through the whole image sequence the appearance and size of the target person in the images change a lot. Here with the proposed method the template is updated robustly and the person is correctly tracked. The template's size is scaled due to the distance changes between the person and the camera.

is still accurate compared with the simple color objects such as Fig.6.4 or Fig.6.7. In

Figure 6.11: The template matching result when two persons are moving in the cluttered outdoor environment. The two persons are moving around each other and one of them is tracked. Here the tracker is lost when the person is occluded by another one (Fig.(6.11.2)). With the proposed Bayesian fusion based algorithm, when the occlusion is finished, the template is updated together with the first template and after the occlusion, the previously tracked person is still correctly tracked.

some situations when there are similar colored regions around the object, the GMMs matching may fail. However, the SSE matching remembers the template's spatial and shape information. Combined SSE with GMMs, it is thus possible to solve the matching problem in this case.

Compared with other methods the Bayesian probability fusion based algorithm gives good results in different situations. For example in the image sequence of Fig.6.10 there are about 45 image frames. With the proposed Bayesian fusion based method in only about 5 frames the tracker is lost. If without template updating, there are about 20 frames with the lost tracker. With the "naive template updating" there are also about 16 frames of lost tracking. With the algorithm of [133] there are about 11 frames of tracking lost. Also the algorithm from [133] or the algorithms from SSE, GMMs template matching can not deal with the situations of Fig.6.7, Fig.6.9 and Fig.6.10. In these three cases the tracked object has large shape deformations. The full comparison of different methods can be seen in Table.6.1.

Image Sequence Name	Total Image Frames Number	Image Frames with Lost Target Tracking				The Tracking Accuracies of the Proposed Method
		Template No Updating	Template “Naive” Updating	Template Updating from [133]	The Proposed Algorithm in this Chapter	
Moving Object Being Occluded Image Sequence (Fig.6.3)	30	7	6	5	2	93.33%
Stationary Object Being Occluded Image Sequence (Fig.6.4)	30	8	7	7	4	86.7%
Object Moving Towards Cameras Image Sequence (Fig.6.6)	50	26	17	9	3	94%
Object with Large Shape Deformations Image Sequence (Fig.6.7)	45	35	27	20	6	86.7%
Object Leaving Image Scene Image Sequence (Fig.6.9)	30	20	14	10	3	90%
Person Moving Towards Moving Cameras Image Sequence (Fig.6.10)	45	20	16	11	5	88.89%
Two Persons Occluding Image Sequence (Fig.6.11)	45	20	16	10	4	91.11%

Table 6.1: The comparison results of different template matching and updating methods. The proposed algorithm can achieve an average accuracy above 90% for the target tracking in different situations.

From the experimental testing, the new template updating algorithm has satisfied two qualities: simplicity and robustness. Simplicity implies that the algorithm is easy to implement (the basic Bayesian and Gaussian Distribution theories) and has the minimum number of parameters (only the warp parameter p and matching PDFs' mean and variance in Eq.6.20 for the matching estimation). Robustness implies the ability of the algorithm to track objects under difficult conditions which include: 1. severe occlusions (Fig.6.3, Fig.6.4 and Fig.6.11). 2. changing of object shape appearance (Fig.6.7 and Fig.6.9). 3. background clutter and the presence of other moving objects in the scene (Fig.6.11). 4. a moving camera (Fig.6.10). 5. non translational object motion like zooms and rotations (Fig.6.6, Fig.6.7, Fig.6.9, Fig.6.10 and Fig.6.11).

6.4 Summary

In this Chapter the normally used SSD (correlation) based template matching method is presented first. The basic theory and the advantages and disadvantages of this method are discussed. The correlation based template matching is used to match the template's optical flow vectors field to the image's clustered optical flow vectors field. This matching is based on the motion vectors, which makes the target features 2D localization in the images independent of its other specific visual properties and also more robust.

Since the SSD (correlation) based method is quite simple and there are several problems, the algorithm with the Bayesian probability based fusion is proposed to improve the template matching performance. In the proposed method the SSE template matching is first selected as the basic matching method. Then the GMMs is combined with SSE to solve the problems of the template's appearance changes and the object occlusions. In this Chapter with the Bayesian fusion of two different template matching methods, a robust template updating method is also proposed. Here the template is updated together with its GMMs parameters and the matching probability PDF parameters to make robust template matching.

In this Chapter the SSE and GMMs methods are fused together with the assumption that they are independent. However in fact these two methods are statistically not inde-

pendent. They both use the template intensity values and are conditioned on almost the same data when trained. Fusing their outputs with an empirically determined weighting factor may overlook this fact. However in this Chapter, for the sake of template modelling, it can be just assumed that they are independent. This simplifying modelling assumption may have some drawbacks. But from the experimental results the template can be accurately matched and tracked, so it is suitable to keep this assumption.

Through the experimental results the fusion of two different template matching methods solve the template matching problems such as the template drifts, shape rotations, appearance changes or occlusions. The average tracking accuracy of the proposed Bayesian fusion based method in different situations is above 90%. Since the algorithm can make accurate template matching and target tracking when the cameras are also moving (Fig.6.10), the potential application of the proposed algorithm can be used for the vision-based mobile robot navigation [43].

The research work in this Chapter improves the existing template matching methods, which makes contributions to the computer vision research.

Updating the template dramatically increases the computational cost of matching; and when updating the template every frame, it is hard to do the real-time image processing and target tracking. In the experimental test in this Chapter, the computational time for each frame using the Bayesian fusion based algorithm is about 2 – 3 minutes based on the Matlab programming. In fact the image template can be just updated whenever necessary instead of updating it every frame, thereby reducing the average computational cost. In this Chapter the template updating with a preset threshold has been developed to solve the problem.

Chapter 7

Target Tracking System

Once the target's $3D$ visual features are identified in the image scene, it is then necessary to estimate the target's $3D$ velocity and position in the world coordinate for autonomous vehicle target tracking. This chapter first presents the sensor data fusion based target's $3D$ velocity and position estimation procedures, which is the fourth step of the proposed visual target tracking scheme. Then as the final step, the Extended Kalman Filter is carried out for the target dynamics estimation. Here the IMM approach and OOSM solution are developed with Extended Kalman Filter to improve the tracking performance.

7.1 Sensor Data Fusion based Target $3D$ Dynamics Information Estimation

In this thesis, the target's $3D$ position and velocity in the world coordinate are estimated for the visual target tracking system. In the previous autonomous vehicle target tracking systems, the target's relative dynamic information from CCD cameras is used directly for the navigation purpose. There are several drawbacks using the relative information. First, these data are only relative from the vision sensors to the target, which are not available for other applications in the same environment. Secondly, in such a situation the vehicle's self rotational and translational motion in the world coordinate are always

not considered, which makes the target dynamics estimation less accurate. So in the proposed visual target tracking scheme in order to make more general applications and get more accurate estimation results than other methods like [34] [43] [223], the target's 3D velocity and position are estimated in the world coordinate for tracking.

7.1.1:

7.1.2:

Figure 7.1: Experiment Setup with the Target's 3D World Coordinate Velocity Estimation Procedures

Let's first consider the cameras mounted on an autonomous vehicle moving in the 3D world coordinate as shown in Fig.7.1.1. Estimation of the target's velocity can be achieved using the algorithm explained in Fig.7.1.2. Here, the motion of the target in the image stream is obtained as optical flow vectors $\vec{V}_T^{O.F.}$. Then $\vec{V}_T^{O.F.}$ is transformed into the image coordinate frame as $\vec{V}_T^{IM}$. Next $\vec{V}_T^{IM}$ is transformed into the camera frame to get the target velocity relative to the cameras $\vec{V}_T^C$.

Here because the cameras are moving in the world coordinate, its axis has the rotational motion R_C^W and translational motion T_C^W relative to the world coordinate. The velocity of the moving cameras in the world coordinate $\vec{V}_C^W$ (consists of R_C^W and T_C^W) can be estimated based on the kinematic model of the autonomous vehicle. Then the target's velocity in the world coordinate $\vec{V}_T^W$ can be estimated as:

$$\vec{V}_T^W = R_C^W \cdot \vec{V}_T^C + T_C^W \quad (7.1)$$

This equation shows how the visual features can be fused with the cameras' motion parameters for estimating the target's velocity.

Following section first explains how $\vec{V}_T^C$ is estimated. Here, the 2D visual cues are fused with the disparity information using a pinhole camera projection model (Fig.7.2) with Eq.7.2 to obtain the target's 3D velocity relative to the cameras.

Figure 7.2: The Pinhole Camera Projection Model

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (7.2)$$

The time derivation of Eq.7.2 gives the velocity.

$$X = \frac{Z}{f}x \Rightarrow \dot{X} = \frac{\dot{x} \cdot Z + \dot{Z} \cdot x}{f} \quad (7.3)$$

where $\dot{X}$ is the target's velocity relative to the camera frame known as $\vec{V}_T^C$ and $\dot{x}$ is the target's image pixel velocity $\vec{V}_T^{IM}$, which can be estimated from the optical flow vectors $\vec{V}_T^{O.F.}$ (Eq.7.4).

$$\vec{V}_T^{IM} = S \cdot \vec{V}_T^{O.F.} \quad (7.4)$$

where S is the effective size of the pixels in the image. In Eq.7.3 Z is the target depth information, which can be estimated from the disparity d (Eq.4.35). The deviation of the depth $\dot{Z}$ is obtained as $\dot{Z} = -b_1 f \frac{\Delta d}{d^2}$, Δd is the difference of the disparity d between the time intervals. From Eq.7.2 and Eq.7.4 $\vec{V}_T^C$ can be obtained as in Eq.7.5

$$\vec{V}_T^C = \begin{pmatrix} V_{T_x}^C \\ V_{T_y}^C \\ V_{T_z}^C \end{pmatrix} = \begin{pmatrix} \frac{1}{f}(\dot{Z} \cdot x + V_{T_x}^{IM} \cdot Z) \\ \frac{1}{f}(\dot{Z} \cdot y + V_{T_y}^{IM} \cdot Z) \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} \frac{1}{f}(-b_1 f \frac{\Delta d}{d^2} \cdot x + S_x \cdot V_{T_x}^{O.F.} \cdot b_1 \frac{f}{d}) \\ \frac{1}{f}(-b_1 f \frac{\Delta d}{d^2} \cdot y + S_y \cdot V_{T_y}^{O.F.} \cdot b_1 \frac{f}{d}) \\ -b_1 f \frac{\Delta d}{d^2} \end{pmatrix} \quad (7.5)$$

where (x, y) is the target 2D image coordinate, which can be estimated from the previous ROI identification.

In order to get the cameras' rotation and translation matrix R_C^W and T_C^W , the cameras' movement in the world coordinate should be known. In this thesis a simple two wheels kinematic model of the autonomous vehicle is used for the experimental testing (Fig.7.3). Two tachometers sensors measure the wheels velocities v_1 and v_2 . Here u_1, u_2 are the

Figure 7.3: Kinemics Model of Autonomous Vehicle

tachometers' output voltages, and k_T is the linear parameter between the tachometers'

output voltages and the wheels velocities, which gives the simple relationship $v_1 = k_T \cdot u_1$ and $v_2 = k_T \cdot u_2$. In order to simplify the tracking system, in this paper it is just assumed that the wheels' slippage is ignored, because in fact more complicated models considering the wheels' slippage effects can only give similar results as the simple models. Then the vehicle's velocity along X and Z directions in Fig.7.3 can be estimated as Eq.(7.6).

$$\begin{pmatrix} v_X \\ v_Z \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{(v_1+v_2)\cos(\theta)}{2} \\ \frac{(v_1+v_2)\sin(\theta)}{2} \\ \frac{v_2-v_1}{2b} \end{pmatrix} = \begin{pmatrix} \frac{(k_T \cdot u_1 + k_T \cdot u_2)\cos(\theta)}{2} \\ \frac{(k_T \cdot u_1 + k_T \cdot u_2)\sin(\theta)}{2} \\ \frac{k_T \cdot u_2 - k_T \cdot u_1}{2b} \end{pmatrix} \quad (7.6)$$

where b is the distance between two wheels and θ is the angle between the X axis and the vehicle's velocity in Fig.7.3.

In this experiment, the velocities in the X and Z directions (Fig.7.1.1) are considered as the most effective movements of the vehicle. The vehicle's velocity can be estimated in all the 3D directions, but this will make the whole system much more complex. So in the thesis, it is just assumed that the vehicle's velocity in the Y -coordinate is too small compared with the other two directions and it can be ignored. The target's position and velocity is still estimated in the 3D world coordinate, only obtained from the moving vehicle without considering the vertical motion. In fact with this assumption, the experimental results are still good.

Based on the above assumption, the camera coordinate has only the Y axis rotation relative to the world coordinate and the rotational velocities in the other two directions can be just assumed very small and neglected. Thus, the whole computation is simplified in a reasonable way. Let θ denote the angle between the camera velocity and X direction, then the camera axis rotation matrix R_C^W can be estimated as:

$$R_C^W = \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix}; \quad \theta_k = \theta_{k-1} + \Delta t \cdot \dot{\theta} \quad (7.7)$$

Here Δt is the time interval from the time $k-1$ to time k to estimate θ from $\dot{\theta}$ in Eq.7.6. In the following equations, the time k for θ is just omitted because all the estimations are

done for the same time stamp k . Now with the kinematic model of the vehicles, problems involving the cameras' 3D motion can be solved for this target tracking system.

At the same time the vehicle's translational vector in the world coordinate can be estimated as T_C^W . Because the vehicle only has velocities in the X and Z directions, based on the autonomous vehicle's kinematic model, T_C^W can be written as Eq.(7.8):

$$T_C^W = \begin{pmatrix} T_C^W x \\ T_C^W y \\ T_C^W z \end{pmatrix} = \begin{pmatrix} \frac{(k_T \cdot u_1 + k_T \cdot u_2) \cos(\theta)}{2} \\ 0 \\ \frac{(k_T \cdot u_1 + k_T \cdot u_2) \sin(\theta)}{2} \end{pmatrix} \quad (7.8)$$

Finally from Eq.(7.1), (7.5), (7.7) and (7.8), the target's 3D velocity $\vec{V}_T^W$ in the world

coordinate can be estimated as $\begin{pmatrix} V_T^W x \\ V_T^W y \\ V_T^W z \end{pmatrix} =$

$$\begin{pmatrix} \frac{1}{f}(-b_1 f \frac{\Delta d}{d^2} \cdot x + S_x \cdot V_T^{O.F.} \cdot b_1 \frac{f}{d}) \cdot \cos(\theta) - b_1 f \frac{\Delta d}{d^2} \cdot \sin(\theta) + \frac{(k_T \cdot u_1 + k_T \cdot u_2) \cos(\theta)}{2} \\ \frac{1}{f}(-b_1 f \frac{\Delta d}{d^2} \cdot y + S_y \cdot V_T^{O.F.} \cdot b_1 f \frac{1}{d}) \\ \frac{1}{f}(-b_1 f \frac{\Delta d}{d^2} \cdot x + S_x \cdot V_T^{O.F.} \cdot b_1 \frac{f}{d}) \cdot \sin(\theta) + b_1 f \frac{\Delta d}{d^2} \cdot \cos(\theta) + \frac{(k_T \cdot u_1 + k_T \cdot u_2) \sin(\theta)}{2} \end{pmatrix} \quad (7.9)$$

The target's 3D position in the world coordinate $\vec{P}_T^W$ can also be obtained from Eq.(7.10).

$$\vec{P}_T^W = R_C^W \cdot \vec{P}_T^C + T_C^W \quad (7.10)$$

Here R_C^W is the camera axis's rotational matrix (Eq.7.7). $\vec{P}_T^C$ is the target position relative to the cameras in the camera frame. T_C^W is the translational vector between the

camera coordinate and the world coordinate (Eq.7.8). So $\vec{P}_T^W$ can be written as:

$$\begin{aligned}\vec{P}_T^C &= \begin{pmatrix} P_{T_x}^C \\ P_{T_y}^C \\ P_{T_z}^C \end{pmatrix} = \begin{pmatrix} \frac{S_x \cdot x \cdot Z}{f} \\ \frac{S_y \cdot y \cdot Z}{f} \\ Z \end{pmatrix} \\ \vec{P}_T^W &= \begin{pmatrix} \cos(\theta) \cdot \frac{S_x \cdot x \cdot Z}{f} - \sin(\theta) \cdot Z + t \cdot \frac{(k_T \cdot u_1 + k_T \cdot u_2) \cos(\theta)}{2} \\ \frac{S_y \cdot y \cdot Z}{f} \\ \sin(\theta) \cdot \frac{S_x \cdot x \cdot Z}{f} + \cos(\theta) \cdot Z + t \cdot \frac{(k_T \cdot u_1 + k_T \cdot u_2) \sin(\theta)}{2} \end{pmatrix}\end{aligned}\quad (7.11)$$

These equations (Eq.7.9 and Eq.7.11) show the data fusion of the target's optical flow vectors ($V_{T_x}^{O.F.}$, $V_{T_y}^{O.F.}$), visual disparity d , image coordinates (x, y) and motion sensors data (u_1, u_2) into one equation to estimate the target's 3D velocity and position in the world coordinate. These two equations are used later to obtain the Extended Kalman Filter measurement function for estimating the target dynamics, which is the proposed data fusion scheme to track the target dynamic states.

7.2 Extended Kalman Filter based Target Tracking System

At this stage, the position and velocity of the target are estimated together using a Kalman Filter. Several other tracking filters can also be used here such as particle filters in [74]. Here Kalman Filter is chosen because of its fast estimation speed and easy implementation. Since the velocity and position measurement models of the target are nonlinear, the derived Kalman Filter will always contain nonlinear components. Therefore, an Extended Kalman Filter (EKF) is used to cater for the non-linear properties. Let X denotes the target's dynamic state, $X = \begin{pmatrix} \vec{P} \\ \vec{V} \end{pmatrix}$. Then the state equation for its movement can be written as Eq.(7.12).

$$X_{k+1} = \Phi_r X_k + v_k \quad (7.12)$$

where Φ_r denotes the target dynamic models, $\vec{P} = (P_x, P_y, P_z)$ denotes the 3D position of the tracked point, and $\vec{V} = (V_x, V_y, V_z)$ denotes its velocity. v_k is the Gaussian white noise. v_k has a 6×6 covariance matrix and over the increment from the time k to time $k - 1$ this process noise enters into both the position and velocity states.

Next, the coordinates of the tracked object in the 2D image can be obtained by (7.13)

$$\begin{pmatrix} P_x \\ P_y \end{pmatrix}_k = \Pi(\vec{P}_k) \quad (7.13)$$

In this equation Π denotes the transformation of a 3D point into the 2D image point (P_x, P_y) and its visual depth P_z .

The resulting measurement vector $\vec{Z}$ for the system can be obtained as given in Eq.(7.14)

$$\vec{Z}_k = H(k, X_k)_k + w_k \quad \vec{Z}_k = \begin{Bmatrix} V_{O.F.x} \\ V_{O.F.y} \\ u_1 \\ u_2 \\ d \\ x \\ y \end{Bmatrix} \quad (7.14)$$

$\vec{Z}_k$ represents the measurements of the target from different sensors. H_k is the measurement function, which can be obtained from Eq.7.9 and Eq.7.11. It is worth noting that $\vec{Z}_k$ comprises of the optical flow vectors $(V_T^{O.F.x}, V_T^{O.F.y})$, disparity d , tachometers sensors data (u_1, u_2) and image spatial information (x, y) into one vector for target tracking. w_k is the measurement noise, which has a covariance matrix of 7×7 . This equation (Eq.7.14) provides a scheme to fuse these different visual cues with the cameras' motion parameters as the inputs to EKF for tracking the target's 3D dynamics as the outputs from EKF.

In the next two sections, the IMM approach and OOSM solution are described together with the Extended Kalman Filter to improve the target 3D dynamics estimation performance.

7.3 IMM Approach for Target Tracking

7.3.1 Introduction

In visual target tracking, the dynamic information aids the tracking process in the presence of occlusions and measurement noises [84]. However, the tracking is complicated by the fact that natural moving targets do not exhibit just one type of motion but complex, unknown and highly nonlinear and time-varying dynamics. The single model approaches do not make full use of the information available and often rely on somewhat ad hoc methods of incorporating uncertainty about the mode estimates (like adjusting the filter noise covariance in proportion to the variance in the model estimate [89]), so the traditional single target linear dynamic models are not quite applicable. Tracking of such targets thus falls into the area of adaptive state estimation (multiple models) [48] [88].

The distinguishing feature of the multiple models approaches is a stochastic (typically Markov) switching process between the models and the simultaneous filtering of the observations through all of the possible systems. The final state estimate is obtained as a weighted combination of the outputs of the estimates from each different model. This is in contrast to single model algorithms which tend to use one model at any given time.

Currently Forsyth et. al. [54] review the linear dynamic models for target tracking. These dynamic models are quite simple and easy to implement and they cover all the target's linear dynamic properties. Thus in this thesis these simple linear dynamic models are combined together with the multiple models algorithm to represent the target's complex motion properties.

Now some work have been done in target tracking using the Multiple Models (MM) filtering [9]. Bar-Shalom and Li [9] review the models in which both the state dynamics and the output matrices switch, and where the switching follows the Markovian dynamics (IMM). They have also presented several different methods for approximately solving the state-estimation problem in the switching models. Some other multiple models tracking systems use the Bayesian network based multi-model fusion methods ([50] and [124]). More recently, switching linear dynamic system (SLDS) models have been studied in [16].

SLDS models and their equivalents have also been studied in the statistics, time-series modelling, and target tracking since early 1970's in [161]. Here it is concluded by Bar-Shalom [9] that the IMM approach performs significantly better than the other multiple models methods. The IMM algorithm computes the state estimate under each possible current model using several filters, with each filter using a different combination of the previous model-conditioned estimates (mixed initial condition). The mixed initial condition of IMM can reduce the estimation errors by adjusting the initial condition for each estimate. IMM is also faster in computation than the other multiple models algorithms. For example IMM requires only r filters to operate in parallel while the generalized pseudo-Bayesian (GPB) multiple models algorithm requires r or r^2 to operate in parallel. Furthermore, the IMM has been shown to be able to keep the estimation errors not worse than the raw measurement errors and provide significant improvements (noise reduction).

Thus in this thesis the IMM approach is developed to fuse several simple and basis linear dynamic models for the target dynamics tracking.

Here the interacting multiple linear dynamic models algorithm provides an approximation of the types of the target employed motion, and then the target dynamic states estimation is obtained with the combination of the weighted different models' Extended Kalman Filter estimates, which can greatly improve the tracking performance. The interacting multiple models (IMM) algorithm here also allows the system to minimize the tracking errors by adjusting the gain of the Extended Kalman Filter.

7.3.2 Linear Dynamic Models for Extended Kalman Filter

There are several simple and basic linear dynamic models available for Extended Kalman Filtering to cover the normal dynamic situations of the moving target [54]. Here the dynamic model is achieved by multiplying the state by some known matrix Φ as in the motion state equation (Eq.(7.12)), and then adding a normal random variable of zero mean and known covariance.

1. **Drifting Points Model.** If the dynamic model is

$$\Phi_k = I \quad (7.15)$$

then the point's new position is its old position, plus some white Gaussian noises term. This dynamic model can be commonly used for objects for which no other better dynamic model is known.

2. **Constant Velocity Model.** $\vec{P}$ gives the position and $\vec{V}$ gives the velocity of a point moving with the constant velocity. The constant velocity model is given in Eq.(7.16).

$$X = \begin{pmatrix} \vec{P} \\ \vec{V} \end{pmatrix}, \quad \Phi_k = \begin{pmatrix} I & (\Delta t)I \\ 0 & I \end{pmatrix} \quad (7.16)$$

3. **Constant Acceleration Model.** $\vec{a}$ is the acceleration of a point moving with the constant acceleration. Then the constant acceleration model is given in Eq.(7.17).

$$X = \begin{pmatrix} \vec{P} \\ \vec{V} \\ \vec{a} \end{pmatrix}, \quad \Phi_k = \begin{pmatrix} I & (\Delta t)I & 0 \\ 0 & I & (\Delta t)I \\ 0 & 0 & I \end{pmatrix} \quad (7.17)$$

4. **Periodic Motion Model.** It is assumed that a point is moving on a line with a periodic movement. Then stack the position and velocity into a vector $\vec{u} = (\vec{P}, \vec{V})$ and with the time interval Δt , the motion model can be obtained with a forward Euler method as shown in Eq.(7.18).

$$\vec{u}_k = \vec{u}_{k-1} + \Delta t \frac{du}{dt} = \vec{u}_{k-1} + \Delta t S \vec{u}_{k-1} = \begin{pmatrix} 1 & \Delta t \\ -\Delta t & 1 \end{pmatrix} \vec{u}_{k-1} \quad (7.18)$$

Now with the above 4 simple and basic linear dynamic models for the target state estimation, one cycle of the IMM algorithm is described next [9].

7.3.3 Interacting Multiple Models Approach

In the Interacting Multiple Models approach (IMM), it is assumed that the system obeys one of a finite number of models. A Bayesian framework is used here starting with the prior probabilities of each model being correct. The model, assumed to be in effect throughout the process, is one of r possible models (the system is in one of r modes) as shown $\{\Phi_j\}_{j=1}^r$.

Then the prior probability that Φ_j is correct (the system is in mode j) is shown in Eq.(7.19).

$$P\{\Phi_j|Z^0\} = \mu_j(0) \quad (7.19)$$

where Z^0 is the prior information and $\sum_{j=1}^r \mu_j(0) = 1$.

1. Calculation of the mixing probabilities ($i, j = 1, \dots, r$). The probability that the mode Φ_i was in effect at $k-1$ given that Φ_j is in effect at k conditioned on Z^{k-1} is Eq.(7.20) and Eq.(7.21).

$$\begin{aligned} \mu_{i|j} &= P\{\Phi_i(k-1)|\Phi_j(k), Z^{k-1}\} \\ &= \frac{1}{\bar{c}_j} P\{\Phi_j(k)|\Phi_i(k-1), Z^{k-1}\} P\{\Phi_i(k-1)|Z^{k-1}\} \end{aligned} \quad (7.20)$$

$$\mu_{i|j}(k-1|k-1) = \frac{1}{\bar{c}_j} p_{ij} \mu_i(k-1), \quad \bar{c}_j = \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad (7.21)$$

Here it is assumed that the model jump process is a Markov process (Markov chain) with the known model transition probabilities p_{ij}

$$p_{ij} = P\{\Phi(k) = \Phi_j | \Phi(k-1) = \Phi_i\} \quad (7.22)$$

2. Mixing ($j = 1, \dots, r$). Starting with $\hat{X}^i(k-1|k-1)$ one computes the mixed initial condition for the filter matched to $\Phi_j(k)$ as

$$\hat{X}^{0j}(k-1|k-1) = \sum_{i=1}^r \hat{X}^i(k-1|k-1) \mu_{i|j}(k-1|k-1) \quad (7.23)$$

The covariance corresponding to the above is

$$\begin{aligned}
 P^{0j}(k-1|k-1) &= \sum_{i=1}^r \mu_{i|j}(k-1|k-1) \{P^i(k-1|k-1) + \\
 &\quad [\hat{X}^i(k-1|k-1) - \hat{X}^{0i}(k-1|k-1)] \\
 &\quad \cdot [\hat{X}^i(k-1|k-1) - \hat{X}^{0j}(k-1|k-1)]'\} \quad (7.24)
 \end{aligned}$$

3. Model-matched filtering ($j = 1, \dots, r$). The estimate Eq.(7.23) and covariance Eq.(7.24) are used as the inputs to the filter matched to $\Phi_j(k)$, which uses $z(k)$ to yield $\hat{X}^j(k|k)$ and $P^j(k|k)$. The likelihood functions corresponding to the r filters

$$\Lambda_j(k) = p[z(k)|\Phi_j(k), Z^{k-1}] \quad (7.25)$$

are computed using the mixed initial condition Eq.(7.23) and the associated covariance Eq.(7.24) as:

$$\Lambda_j(k) = p[z(k)|\Phi_j(k), \hat{X}^{0j}(k-1|k-1), P^{0j}(k-1|k-1)] \quad (7.26)$$

4. Model probability update ($j = 1, \dots, r$). This is done as follows:

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j, \quad c = \sum_{j=1}^r \Lambda_j(k) \bar{c}_j \quad (7.27)$$

where $\bar{c}_j$ is the expression from Eq.(7.21).

5. Estimate and Covariance combination. Combination of the model-conditioned estimates and covariances is done according to the mixture equations

$$\hat{X}(k|k) = \sum_{j=1}^r \hat{X}^j(k|k) \mu_j(k) \quad (7.28)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{P^j(k|k) + [\hat{X}^j(k|k) - \hat{X}(k|k)][\hat{X}^j(k|k) - \hat{X}(k|k)]'\} \quad (7.29)$$

Note that this combination is only for output purposes. It is not part of the algorithm

recursions.

7.3.4 Interacting Multiple Models For Extended Kalman Filter Estimation

Recall the basic model equations for the Extended Kalman Filter Eq.(7.30).

$$\begin{aligned} X(k) &= \Phi(k-1) * X(k-1) + v(k-1) \\ Z(k) &= H(k) * X(k) + w(k) \end{aligned} \quad (7.30)$$

where Φ is the state transition matrix, X is the state vector (the $\begin{pmatrix} \vec{P} \\ \vec{V} \end{pmatrix}$ in Eq.(7.12)), v is the process noise, Z is the measurement vector, and $H(k)$ is the measurement function based on the Jacobian computation of $H(k) = \frac{\partial H_k}{\partial X_k}$ from Eq.7.14 for the Extended Kalman Filtering, and w is the measurement noise. The actual state estimation and prediction equations are given as Eq.(7.31):

$$\begin{aligned} X(k|k-1) &= \Phi(k-1) * X(k-1|k-1) \\ X(k|k) &= X(k|k-1) + G(k) * residue(k) \\ residue(k) &= Z(k) - H(k) * X(k|k-1) \\ P(k|k-1) &= \Phi(k-1) * P(k-1|k-1) * \Phi(k-1)^T + Q(k-1) \\ S(k) &= H(k)P(k|k-1)H(k)^T + R(k) \end{aligned} \quad (7.31)$$

where $residue(k)$ is the measurement residue, a Gaussian random variable with the zero mean and covariance $S(k)$, and $R(k)$ is the covariance of the measurement noise w .

The equation for the filter gain, G , and the covariance matrix, P , of the state prediction are then as Eq.(7.32):

$$\begin{aligned} G(k) &= P(k|k-1) * H(k)^T * (H(k) * P(k|k-1) * H(k)^T + R(k))^{-1} \\ P(k|k) &= (I - G(k) * H(k)) * P(k|k-1) \end{aligned} \quad (7.32)$$

where $Q(k)$ is the covariance of the process noise v , $H(k)$ is the measurement matrix

from Eq.(7.30) and $\Phi(k)$ is the state transition matrix from Eq.(7.30).

Here according to the Interacting Multiple Models (IMM) approach, several simple linear dynamic models Φ introduced in the previous sections are used as the different state transition matrix (Eq.7.12) for the Extended Kalman Filtering. In the interacting multiple dynamic models algorithm, the likelihood of each mode-conditioned estimate $\Lambda_j(k)$ as shown in Eq.(7.26) is generated with the Extended Kalman Filter measurement residue $residue_j(k)$ (Eq.(7.31)) as Eq.(7.33).

$$\Lambda_j(k) = N[residue_j(k); 0, S_j(k)] \quad (7.33)$$

Now with the different model's Extended Kalman Filter state estimate X^j together with the covariance estimate P^j and based on the above recursive IMM procedures, the interacting multiple linear dynamic models for the Extended Kalman Filter based target tracking can be achieved.

7.4 The OOSM Problem and Solution for Extended Kalman Filtering

7.4.1 Introduction

Recently the delayed measurements problem for multi-sensor fusion based tracking systems has been theoretically studied and solved by Bar-Shalom in [6] [8]. In normal target tracking systems, different sensors' measurements are typically collected discretely and then processed in the fusion algorithm. Each measurement is provided with a "time-stamp" for the tracking filter estimation. In multi-sensor fusion based tracking systems operating in a centralized manner [6], measurements from each sensor reach the fusion center in different time stamps. There are usually different time delays in transmitting and processing the measurements from various sensors, which results in that the target's measurements arrive out of sequence. When the Out-Of-Sequence Measurement (OOSM) arrives and is associated with the time stamp κ , the system has already been

processed to the time stamp $t > \kappa$. Thus there exists the problem of using the “older” measurement from the time stamp κ to update the state at the time stamp of t . In Bar-Shalom’s paper [6], it is also known as the “negative-time measurement update” problem (because $\kappa - t$ is negative and in the standard filtering problem it is supposed to be nonnegative). The OOSM target tracking problems can also be found in some real multi-sensor systems [214].

Currently some work have been done with the OOSM target tracking problems. Bar-Shalom [6] first presents the exact state update solutions to this problem. In the work of [6] one optimal algorithm together with two approximately suboptimal techniques are compared and discussed to solve the OOSM estimation problem. These methods solve the OOSM estimation problem where the OOSM lies between the two last measurements, i.e., its lag is less than a sampling interval-the $1 - \text{step} - \text{lag}$ case. The real world has, however, OOSMs with the arbitrary lag. Subsequently, the suboptimal algorithm is extended to the case of an arbitrary (multistep) lag, but the resulting algorithm requires a significant amount of storage. Later in the work of [8] they just show how the $1 - \text{step} - \text{lag}$ algorithm can be generalized to handle an arbitrary (multistep) lag (the problem mentioned in [6]) while preserving their main features of solving the update problem without iterating. This is accomplished by defining an equivalent measurement at the current time that represents all the measurements with the time stamps later than the OOSM. This yields algorithms that carry out in one step that updates the current state with the $l - \text{step} - \text{lag}$ OOSM instead of the conventional way of using an algorithm that requires l steps. Their method leads to a very small (a few percent) degradation of MSE performance. Also Challa et. al. [22] formulate and solve the OOSM problem in a Bayesian framework. They establish that the OOSM solution involves the joint probability density of current and past states. Based on this, AS-KF (Augmented State Kalman Filter) is proposed as the fundamental solution to this problem in the linear Gaussian case. Their new augmented state algorithms in [22] handle the process noise-target state cross correlation implicitly and can be readily extended to handle the clutters. AS-PDA is also proposed in [22] to deal with the data association issues arising from the presence of the clutters in the OOSM problem.

Previously Kosaka et. al. [109] propose another kind of system solving the delayed mea-

surement problems for mobile robot navigations. In their method the position updating using vision is retroactive in the sense that the robot does not wait for the results of vision processing to become available. While the computer is engaged in vision calculations, the robot continues to proceed towards its goal. When the vision results do become available, the robot then retroactively updates the positional uncertainties. In fact the scheme proposed in [109] is rather complex. There are also several drawbacks. First, only when the accumulative measurement errors exceed a certain threshold, the delayed measurements are considered and the mobile robot then stops to capture images for replanning the path. It is not a good way to solve the delayed measurement problem. This system can not continuously correct the target state estimates based on the measurements, which limits the whole system's estimation accuracy. Secondly, it also requires a lot of system memory to store the images. Thirdly, the method proposed in [109] is too simple to deal with the target's 3D complex movements. Different from this method, the algorithm proposed in this thesis solves the delayed measurement problem right in the Extended Kalman Filter, which gives continuous corrections to the errors and noises. At any given point in time, the estimates of target's position and velocity incorporate information from all measurements received, even if received out of sequence and the system always gives the best estimates based on the data. Also the OOSM solution from Bar-Shalom is fast in computation. It does not put any more computational burden on Extended Kalman Filter, which is suitable for the real-time testing. So in this section the OOSM solution from Bar-Shalom is developed as the better solution for the sensor fusion based target tracking system.

In this thesis the tachometers sensors and vision sensors have different data capturing rates. Also since the target's 3D visual features are obtained from separate image processing algorithms, the computational time for each measurement is quite different. Even assuming that all the data are arriving within one time interval, there is also the measurements' time discretization problem, which means that the delayed measurements are associated with different time stamps. In such a situation, the fusion algorithm receives the measurement information in different time stamps, which leads to the OOSM problem. In the previous multi-sensor fusion based target tracking systems, the OOSM problem is often ignored or just simply solved by using some human reasoning methods

like [109]. However in this thesis in order to make an accurate target dynamic state estimation, the $1 - \text{step} - \text{lag}$ solution from Bar-Shalom [6] is used for the Extended Kalman Filter to solve this problem.

In this thesis in order to simplify the OOSM problem, the measurement delays from different sensors are assumed to be within one time interval. In such a way, there is no multistep OOSM tracking problem ($l - \text{step} - \text{lag}$ OOSM) but only the $1 - \text{step} - \text{lag}$ OOSM problem [87]. Next the solution from Bar-Shalom [6] is used to estimate the target state and solve the OOSM problem.

In this thesis the IMM approach and OOSM solution are presented together with Extended Kalman Filter to improve the target tracking performance. In fact combining the OOSM solution with the IMM approach is rather complicated. In this thesis the experimental testings are done off-line, so the OOSM solution can be just assumed to be independent of the previous IMM approach and they are applied separately to test the tracking performance. However most recently Bar-Shalom et. al. [7] just present the algorithm for incorporating the OOSM into an IMM estimator. The work comes out in January 2005, after the date when the research work in this thesis have already been done. The IMM-OOSM algorithm in [7] is considered as the future improvement for the proposed visual target tracking scheme.

In the OOSM solution presented next the dynamic model used in Extended Kalman Filter is the constant velocity model.

7.4.2 Formulation of the OOSM Problem

Based on the formulations in Bar-Shalom's [6], the OOSM problem for the Extended Kalman Filter estimation is explained as follows.

First, in the Extended Kalman Filter the system's states and measurements are modelled by

$$X(k) = F(k, k-1)X(k-1) + v(k, k-1); \quad Z(k) = H(k)X(k) + w(k) \quad (7.34)$$

where $F(k, k-1)$ is the state transition matrix from the time t_{k-1} to the time t_k (the nearly constant velocity dynamic model in Eq.7.12) and $v(k, k-1)$ is the process noise for this interval. $x(k)$ is the target state estimated by the Extended Kalman Filter (the $\begin{pmatrix} \vec{P} \\ \vec{V} \end{pmatrix}$ in Eq.7.12). $z(k)$ represents the sensors' measurements estimated from Eq.7.14.

The time τ for the OOSM is within one time interval as:

$$t_{k-l} \leq \tau < t_{k-l+1} \quad (7.35)$$

This is called the *1-step-lag* problem, even though the lag $l=1$ is really a fraction of a time step. The noises in Eq.7.34 are assumed as zero mean, white with covariances

$$E[v(k, j)v(k, j)'] = Q(k, j), \quad E[w(k)w(k)'] = R(k) \quad (7.36)$$

and mutually uncorrelated.

Next, the state equation corresponding to the OOSM time τ is

$$X(k) = F(k, \kappa)X(\kappa) + v(k, \kappa) \quad (7.37)$$

Here κ is the discrete time notation for τ . The above equation can be rewritten backward as

$$X(\kappa) = F(\kappa, k)[X(k) - v(k, \kappa)] \quad (7.38)$$

where $F(\kappa, k) = F(k, \kappa)^{-1}$ is the backward transition matrix.

Then, the OOSM problem is explained as follows. Without the OOSM, the system is updated to $t = t_k$ by the Extended Kalman Filter as (assuming Gaussian noises)

$$\hat{X}(k|k) = E(X(k)|Z^k), \quad P(k|k) = cov(X(k)|Z^k) \quad (7.39)$$

where the cumulative set of measurements at t_k is

$$Z^k = \{z(i)\}_{i=1}^k \quad (7.40)$$

When the earlier measurement from the discrete time κ ,

$$z(\kappa) \approx z(\tau) = H(\kappa)X(\kappa) + w(\kappa) \quad (7.41)$$

arrives after Eq.7.39 has been calculated. The state estimate has to be updated with the earlier measurement from Eq.7.41, namely, to calculate

$$\hat{X}(k|\kappa) = E[X(k)|Z^\kappa], \quad P(k|\kappa) = \text{cov}[X(k)|Z^\kappa] \quad (7.42)$$

where

$$Z^\kappa = \{Z^k, z(\kappa)\} \quad (7.43)$$

7.4.3 Summary of The OOSM Solution Algorithm

As explained earlier the time interval between two estimation states is set to make all the measurements arrive only within one time interval, which transfers the OOSM problem from a multiple-lag OOSM to a simple *1 – step – lag* OOSM [6]. In such a way the estimation can be greatly simplified and the estimation accuracy can also be sustained.

The *1 – step – lag* solution used in this Chapter is proposed by Bar-Shalom in [6], which consists of following steps: 1. The state from the latest time k is retrodicted to the earlier time κ , accounting in full for the process noise $v(k, \kappa)$, the last term in Eq.7.38, which has a non-zero conditional mean given Z^k . 2. Evaluate the corresponding covariance. 3. Calculate the filter gain for updating the state $X(k)$ with the earlier measurement $z(\kappa)$. 4. Update the state estimate $\hat{X}(k|k)$ to $\hat{X}(k|\kappa)$ and calculate the corresponding covariance.

First, the state from k is retrodicted to κ as

$$\hat{X}(\kappa|k) = F(\kappa, k)[\hat{X}(k|k) - Q(k, \kappa)H(k)'S(k)^{-1}v(k)] \quad (7.44)$$

Next, the covariances for the OOSM estimation are calculated sequentially as follows:

- The state retrodiction's covariances

$$P_{vv}(k, \kappa|k) = Q(k, \kappa) - Q(k, \kappa)H(k)'S(k)^{-1}H(k)Q(k, \kappa) \quad (7.45)$$

$$P_{xv}(k, \kappa|k) = Q(k, \kappa) - P(k|k-1) \times H(k)'S(k)^{-1}H(k)Q(k, \kappa) \quad (7.46)$$

$$P(k|k) = P(k|k-1) - P(k|k-1)H(k)'S(k)^{-1} \times H(k)P(k|k-1) \quad (7.47)$$

Here $P_{vv} = \text{cov}\{v(k, \kappa)|Z^k\}$ and $P_{xv} = \text{cov}\{X(k), v(k, \kappa)|Z^k\}$.

- The state retrodiction's covariance

$$P(\kappa|k) = F(\kappa, k)[P(k|k) + P_{vv}(k, \kappa|k) - P_{xv}(k, \kappa|k) - P_{xv}(k, \kappa|k)']F(\kappa, k)' \quad (7.48)$$

- The retrodicted measurement's covariance

$$S(\kappa) = H(\kappa)P(\kappa|k)H(\kappa)' + R(\kappa) \quad (7.49)$$

- The covariance between the state at k and the OOSM measurement at κ

$$P_{xz}(k, \kappa|k) = [P(k|k) - P_{xv}(k, \kappa|k)F(\kappa, k)'H(\kappa)'] \quad (7.50)$$

Then, the gain used for the update is

$$W(k, \kappa) = P_{xz}(k, \kappa|k)S(\kappa)^{-1} \quad (7.51)$$

Finally, the system is update with the OOSM $z(\kappa)$ of the most recent state estimate $\hat{x}(k|k)$ as

$$\hat{X}(k|\kappa) = \hat{X}(k|k) + W(k, \kappa)[z(\kappa) - H(\kappa)\hat{X}(k|k)] \quad (7.52)$$

The covariance of the updated state estimate is

$$P(k|\kappa) = P(k|k) - P_{xz}(k, \kappa|K)S(\kappa)^{-1}P_{xz}(k, \kappa|k)' \quad (7.53)$$

Now with the above procedures, the 1-step-lag OOSM estimation solution can be

achieved for the fusion based Extended Kalman Filtering.

7.5 Summary

In this Chapter, the novel sensor data fusion algorithm is proposed to estimate the target's $3D$ dynamics in the world coordinate. Then an Extended Kalman Filter based tracking scheme is described in details. Next the IMM and $1 - \text{step} - \text{lag}$ OOSM solution algorithms are applied and developed to be integrated with the Extended Kalman Filter to improve the target tracking performance.

The target's final $3D$ world coordinate dynamics tracking performance is demonstrated and evaluated through experiments under different situations in the next Chapter. The discussion commenting the tracking achievements is also presented in the next Chapter.

Chapter 8

Target Tracking Experimental Results

In order to evaluate the performance and tune the parameters of the proposed system, the proposed visual target tracking scheme is tested under different situations. Extensive experimental results are presented in this Chapter.

In this experiment, a moving platform is designed with several sensors (Tachometers sensors, Attitude Heading Reference System (AHRS) and a CCD stereo cameras pair) as shown in Fig.8.1 to simulate an autonomous vehicle. Tachometers sensors measure the speeds of the rear wheels of the platform. AHRS measures the acceleration and velocity heading angle between the two motion directions of the platform. The heading angle and acceleration data can be used as the noise information or acceleration measurement for the tracking filter. Two color CCD cameras (the focus length $f = 5.4mm$) are connected with the Hitachi IP5005 image processing cards for video images capturing and processing. The captured image size is 256×220 pixels. The capturing rate is 15 frames/s. All the sensors are connected with the embedded CPU (low CPU), and then through TCP/IP the low CPU is connected with the industry PC (high CPU) for the data capturing and processing.

The processing of the video images takes a lot of computational time. The images are captured at a $15Hz$ rate, but the optical flow vectors estimation and stereo disparity

8.1.1:

8.1.2:

Figure 8.1: The moving platform with different sensors. In this experiment the moving platform is designed based on a man-push trolley. On this trolley, two PCs (industry PC and embedded PC) and three sensors (AHRS, tachometers and CCD Cameras) are mounted together to do sensors data capturing and processing for the target tracking system.

calculation can not be done at the same time at such a high rate using the current computer hardware. Also in order to simplify the OOSM problem into a *1 – step – lag* problem, the time interval between two consecutive states is assumed to be large enough to make all the measurements arrive within one time interval. All these make the target tracking system difficult to work in real-time, so the following experiments are done off-line. However with the rapid development of the hardware especially the image processing cards (FPGA) available, in the future it is quite reasonable to make this tracking system perform well in real-time.

First the noise identification procedures for the tracking system are described.

8.1 Noise Identification for the Tracking System

The noise identification for a tracking system can be divided into three major categories [9]:

1. Noise Identification for the Continuous State Space Model.
2. Noise Identification for the Discrete State Space Model.
3. Noise Identification for the Direct Discrete Time System.

In this thesis data are obtained in discrete time from different sensors. The tracking model in this thesis is considered as the discrete state space model. So the noise identification procedures for the discrete state space model are described here. The following procedures are taken from [9].

8.1.1 Noise Identification for the Discrete State Space Model

Since, in general, the state observations are done in discrete time, the corresponding discrete-time state equations are needed. The (dynamic) model for the discrete time linear stochastic systems can be written with the simplified index-only time notation as

$$X(k+1) = \Phi(k)X(k) + G(k)u(k) + v(k) \quad (8.1)$$

where the input is assumed known along with the matrices $\Phi(k)$ and $G(k)$ and the process noise $v(k)$ is a zero-mean, white random sequence with the covariance matrix $Q(k)$. Any (known) nonzero mean of the process noise v can be incorporated into the input.

The discrete time measurement equation is, with a similar notation

$$Z(k) = C(k)X(k) + \omega(k) \quad (8.2)$$

where $C(k)$ is the measurement matrix, $\omega(k)$ is the measurement noise-a random sequence with moments:

$$\begin{aligned} E[\omega(k)] &= 0 \\ E[\omega(k)\omega(j)'] &= R(k)\delta_{kj} \end{aligned} \quad (8.3)$$

The process and measurement noise sequences are (usually) assumed uncorrelated, that is

$$E[v(k)\omega(j)'] = 0 \quad \forall k, j \quad (8.4)$$

The discrete time state equation with the sampling period T is

$$X(k+1) = \Phi X(k) + v(k) \quad (8.5)$$

The covariance of the discrete time process noise is $v(k)$ and assuming $\tilde{q}$ to be constant. For the second order models, such as the white noise acceleration state model (the nearly constant velocity model), there is:

$$\begin{aligned} Q &= E[v(k)v(k)'] \\ &= \int_0^T \begin{bmatrix} T-\tau \\ 1 \end{bmatrix} [T-\tau, 1] \tilde{q} d\tau \\ &= \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} \tilde{q} \end{aligned} \quad (8.6)$$

The changes in the velocity over a sampling period T are of the order of

$$\sqrt{Q_{22}} = \sqrt{\tilde{q}T} \quad (8.7)$$

This equation can serve as a guideline for the process noise intensity choice, the choice of the power spectral density $\tilde{q}$ of the process noise in this model. A nearly constant velocity model is obtained by the choice of a “small” intensity $\tilde{q}$ in the following sense: the changes in the velocity have to be small compared to the actual velocity.

If the dynamic model of the system is the third order model (such as the Wiener process acceleration state model (the nearly constant acceleration model),

$$\Phi = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (8.8)$$

and the covariance matrix of $v(k)$ in this situation is then given by:

$$\begin{aligned} Q &= E[v(k)v(k)'] \\ &= \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix} \tilde{q} \end{aligned} \quad (8.9)$$

The changes in the acceleration over a sampling period T are of the order of

$$\sqrt{Q_{33}} = \sqrt{\tilde{q}T} \quad (8.10)$$

This can also serve as a guideline in the process noise intensity choice-the choice of the power spectral density $\tilde{q}$ of the continuous time process noise $\tilde{v}$ for “tuning” this model to the actual motion of the object of interest. A nearly constant acceleration model is obtained by choosing a “small” intensity $\tilde{q}$ in the following sense: the changes in the acceleration should be small relative to the actual acceleration levels.

8.1.2 Measurement Noise

The basic idea to identify the measurement noises is to do some sensor testing before the experiments. From the testing, the error orders can be defined. In the experiments, the measurements noise can be assumed to be the normal zero-mean Gaussian white noise. The noise order can be decided with the previous experimental data trial and error.

8.1.3 Noise Initialization of the State Estimator

A state estimation filter is called consistent if its estimation errors are “commensurate” or “compatible” with the filter-calculated covariances. At initialization, it is just as important that the covariance associated with the initial estimate reflects the real accuracy.

According to the Bayesian model, the true initial state is a random variable, assumed to

be normally distributed with a known mean-the initial estimate-and a given covariance matrix

$$X(0) \sim N[\hat{X}(0|0), P(0|0)] \quad (8.11)$$

The norm (“chi-square”) test for the initial estimation error is

$$\tilde{X}(0|0)'P(0|0)^{-1}\tilde{X}(0|0) \leq c_1 \quad (8.12)$$

where c_1 is the upper limit of the, say, 95% confidence region from the chi-square distribution with the corresponding number of degrees of freedom.

Sometimes one has to choose the initial covariance. The “choice” has to be such that Eq.(8.12) is satisfied. A large error in the initial estimate, if the latter is deemed highly accurate, will persist a long time because it leads to a low filter gain and thus the new information from the measurements receives weighting that is too low.

For the one-dimensional case Eq.(8.12) can be stated as follows: The initial error should be not more than two times the associated standard deviation, which is called a “ 2σ ” error. If the initial variance is such that the initial error is “ 1σ ” and the subsequent measurements are accurate, then the initial error will decrease rapidly [9].

8.1.4 Root-Mean-Square Error

The model estimation performance can be evaluated by the Root-Mean-Square Error (RMSE).

$$RMSE_m = \sqrt{\frac{1}{N} \sum_{i=0}^N [\hat{F}_m(z_i) - F_m(z_i)]^2} \quad (8.13)$$

where the subscript m indicates the m – th profile. When a group of profiles are considered, the mean value can be computed,

$$RMSE = \frac{1}{M} \sum_{m=1}^M RMSE_m \quad (8.14)$$

where M is the total number of profiles. If the model is perfect, $(\hat{F}_m(z_i) - F_m(z_i))$, $RMSE_m$ should be zero. Usually, the model is not perfect. The smaller the RMSE is, the better the performance of the model is.

Now after all the noises are identified, the tracking system is tested in both the indoor structured and outdoor unstructured situations.

8.2 Experimental Results in the Structured Environment

Figure 8.2: The image sequence of a moving target with the moving background in a structured indoor environment. A blue color object is moving from left to right in the image sequence. It can be seen that the background is changing frame by frame.

As shown in Fig.(8.2) the platform carrying the cameras is moving towards the moving target, and as a result the size of the object appearing in the scene increases from left to right. As explained earlier, the tracking system, proposed in this thesis, can handle the changes in the visual depth as well as the varying background conditions.

This image sequence in the indoor structured environment is processed with the image processing steps of visual features extractions (optical flow vectors and 3D disparities estimation), image segmentation and clustering and template matching. The step by step results can be seen from the previous chapters' experimental results sections (optical flow estimation results from Chapter 4, image segmentation and clustering results from Chapter 5 and template matching results from Chapter 6).

In these experiments the stereo images are captured as the vision sensor data. The image processing is done only with one of the stereo images sequences and the 2D

visual features are estimated in one image sequence. The stereo images are only used to estimate the disparity depth. In such a way, the double image processing is avoided, which can reduce the total image processing cost.

At the same time the target's 3D depth is estimated from the stereo images based on the image disparities. Here the disparity is estimated only for the previously extracted

8.3.1:

8.3.2:

Figure 8.3: The stereo images and distance estimation results. One pair of stereo images are shown in Fig.(8.3). Fig.(8.3.2) shows the estimated visual depth vs. the frame number. In Fig.(8.3.2) the x axis is the image frame number. The y axis is the estimated distance (cm) between the cameras and the object.

region of interest (ROI). In such a way the whole image matching is avoided and the system computational time can be reduced. It can be seen from Fig.8.3.2 that the distance between the target and cameras are decreasing as the platform moves towards the target. The estimated depth results are compared with the measured distances (ground true data). The estimation results are satisfactory.

In the indoor laboratory experiments, a controlled environment is built for testing. The cameras platform's movement is measured and marked on the ground, so the ground true data like the depth and the following moving platform's trajectory can be obtained.

While capturing the images, the two tachometer sensors measure the motion parame-

ters of the moving platform. Fig.(8.4.1) shows the moving platform's velocity in the 2D

8.4.1:

8.4.2:

Figure 8.4: The motion sensors data and cameras' platform movement trajectory in the 2D plain. Fig.(8.4.1) shows the estimated two wheels velocities. Here the x axis is the frame number. The y axis is the velocity (cm/s). From the estimated velocity of the moving platform, the platform's movement trajectory can be plotted as shown in Fig.(8.4.2). The x and y axes are the 2D positions in the plain (cm).

(X, Z) plain (Fig.(7.1.1)). The platform's velocity is estimated using the two tachometers sensors' readings based on the vehicle's kinematic model. Fig.(8.4.2) presents the estimated trajectory of the moving platform compared with the ground truth trajectory in the 2D plain, which also agrees that the cameras are moving towards the target. The difference between the estimated trajectory and the ground truth data is not large (only centimeters). This reflects that the tachometers sensors can give the satisfactory moving platform's velocity estimation results for the indoor experimental testing.

Now the cameras' velocity is fused with the optical flow vectors and visual disparity values for the target's 3D position and velocity estimation in the world coordinate. The velocity of the target in the world coordinate frame is estimated based on Eq.(7.9). After the target's 3D world velocity and position is estimated, the Extended Kalman Filter is carried out for target tracking.

The vector $(\vec{P}, \vec{V})^T$ in Eq.7.12 is rewritten as $(p_x, v_x, p_y, v_y, p_z, v_z)^T$ to make the tracking

estimation more easy to be implemented, where (p_x, p_y, p_z) represents the target's 3D position and (v_x, v_y, v_z) represents the target's 3D velocity. Then the state estimation noise $v(k, k-1)$ in Eq.7.12 can be considered as zero mean, white with covariance, which is obtained based on the previous noise identification procedures from [9]. The noise in each coordinate axis is assumed independent, then it has:

$$Q = E[v(k, k-1)v(k, k-1)'] = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_1 & 0 \\ 0 & 0 & Q_1 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t \end{bmatrix} \tilde{q} \quad (8.15)$$

where Δt is the sampling interval. Q_1 is the process noise covariance matrix component for each direction of the target's 3D movements. The choice of the power spectral density $\tilde{q}$ of the process noise is selected based on the previous experimental testings. Here the process noise variance matrix for the EKF is taken as $Q_1 = \begin{pmatrix} 1e-3 & 0 \\ 0 & 1e-3 \end{pmatrix}$.

The measurement noise $w(k)$ in Eq.7.14 is independent of v , and zero mean, white with covariance

$$E[w(k)w(k)^T] = R \quad (8.16)$$

where R is the measurement noise covariance, which is a 7×7 matrix. R is selected based on the different sensors' characteristics and also the optical flow vectors estimation, stereo disparity calculation and all the other image processing's accuracies. In this thesis, the disparity and displacement fields are computed for each time step and superimposed with the additive Gaussian white noise with a variance 1/12.

The estimation starts with the initial covariance $P(1|1)$ (a 6×6 matrix) of the state at $k = 1$ (based on the 2-point differentiation)

$$P(1|1) = \begin{bmatrix} P_{11} & 0 & 0 \\ 0 & P_{11} & 0 \\ 0 & 0 & P_{11} \end{bmatrix}, \quad P_{11} = \begin{bmatrix} r & r/\Delta t \\ r/\Delta t & 2r/\Delta t^2 \end{bmatrix} \quad (8.17)$$

and proceeds to $k = 2$. P_{11} is the covariance matrix component for each direction of the target's 3D movements. r can be normally set based on the measurement noise $w(k)$

and its covariance R .

Here in order to show the performance of the target's 3D tracking in a reasonably easy and clear way, a white rectangle is plotted to represent the target's tracked region in the 2D images. This rectangle is obtained by projecting the 3D position of the target in the world coordinate frame onto the 2D image using a pinhole camera projection model (Eq.7.13). If the target's 2D position can be precisely localized through the image sequence, then the target's 3D world coordinate dynamics is accurately tracked.

Figure 8.5: The target's 3D tracking results without template updating in the structured indoor environment. The target tracking results are not good in these figures.

Fig.(8.5) shows the tracking result without template updating. The sizes of the target's region in different images are the same. From these images, the tracking results can not reflect the exact shape of the object.

Figure 8.6: The target tracking results with template updating in the structured indoor environment. The target's 3D position is projected into the 2D images to show the performance. The IMM method is implemented with the Extended Kalman Filter to improve the tracking performance. The size of the template is changing due to the distance changes. The rectangle can accurately fit the target region which shows the tracking system performs well.

Fig.(8.6) shows the 3D target tracking results with the updating template. The target tracking is done based on the Extended Kalman Filtering with the IMM estimation.

Figure 8.7: The target position tracking results in the 2D images. The target's 3D position is projected into the 2D images to show the performance. In Fig.8.7 the measurement target 2D position and the estimated target 2D position along the trail are compared.

Fig.(8.7) shows the projected horizontal position of the target in the image along the x-axis of the plot and the estimated height of the target in the image along the y-axis. As the cameras are moving closer to the target, the target appears bigger and bigger in the images from left to right. The “o –” lines are the target's marked positions along the trail (ground true data). In the indoor laboratory experiments the target's movements along the trail are marked and in such a way the ground truth measurements are obtained. The “+ –” lines are the target's estimated positions from the proposed algorithm. The differences between them are not large (only several centimeters). When the estimated 3D positions are projected into the 2D images as shown in Fig.8.6.1, the target is precisely localized and the differences can be just neglected. It is clear from the figures that the proposed scheme tracks the 3D position of the target accurately in the world coordinate.

In this thesis for the IMM based Extended Kalman Filtering, the Markov chain transition matrixes between different linear dynamic models (Eq.(7.22)) are taken from [9]. The matrix for the second order models (Drifting Points Model, Constant Velocity Model

and Periodic Motion Model) is:

$$[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \quad (8.18)$$

The final results are not very sensitive to these values (e.g., p_{11} can be set between 0.8 and 0.98). For the three order model (Constant Acceleration Model) the Markov chain transition matrix can be

$$[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 & 0 \\ 0.33 & 0.34 & 0.33 \\ 0 & 0.05 & 0.95 \end{bmatrix} \quad (8.19)$$

As explain early in the previous Chapter, the combination of IMM and OOSM is not considered in this thesis, so the IMM and OOSM solution algorithms are tested separately. Thus in the following estimation figures, the position and velocity estimates from the IMM and OOSM algorithms are in different ranges and they are not corresponding to the same time states, because they are obtained from different set of experiments.

The results of the target's first 20 states are shown to demonstrate the estimation performance with the IMM algorithm. From Fig.8.8 the target has quite different dynamic motion properties. In the target velocity estimation figure (Fig.8.8.2), the target's velocity is hardly following any simple linear dynamic model. The velocity is changing continuously. In these figures the estimates from the four simple linear dynamic models and IMM are compared together with the measurements (ground truth data). It is clear that the IMM estimates are the best among all the dynamic models. Some models like the periodic model even diverge very fast. The RMS errors are based on the multiple repetitions of experiments. The RMS error is a better way to demonstrate the tracking performance and show better comparison results. Compared with other models the RMS errors from the IMM estimation are also relatively smaller with the smallest peak error.

Fig.8.10 shows the estimation results when the target moves with a direction turn. In such a situation, other linear dynamic models poorly track the target since the velocity changes too much. There are large estimation lag errors at the motion direction turn point. The periodic model even diverges very fast. However the IMM algorithm still

8.8.1:

8.8.2:

Figure 8.8: Target's position and velocity estimation results from different models. The estimates along the x dimension of the target's 3D world coordinate position and velocity are shown here. The estimation results with the IMM algorithm are relatively more accurate than those from other models. In this figures the target position measurements are the ground truth values.

8.9.1:

8.9.2:

Figure 8.9: Multiple repetitions of the experiments are performed and the RMS errors are compared. The estimates along the x dimension of the target's 3D world coordinate position and velocity are shown here. Here the RMS errors from the estimation results with the IMM algorithm are relatively smaller.

gives quite accurate position estimates. Without the maneuver detection decision, the IMM algorithm is able to accurately track the target motion with the direction turn.

8.10.1:

8.10.2:

Figure 8.10: Target dynamic information estimation results from different models when the target is moving with a direction turn. The estimates along the x dimension of the target's 3D world coordinate position and velocity are shown here. In such a situation, the estimation results with the IMM algorithm are still quite accurate. In these figures the target position measurements are the ground truth values.

8.11.1:

8.11.2:

Figure 8.11: Multiple repetitions of the experiments are performed and the RMS errors are compared. The estimates along the x dimension of the target's 3D world coordinate position and velocity are shown here. Here the RMS errors from the estimation results with the IMM algorithm are relatively smaller.

In the proposed tracking system the tachometers sensors and vision sensors have different data capturing rates. Also since the target 3D visual features need to be obtained

from different image processing algorithms, the computational time for each method is different and usually quite long. In such a way, the fusion algorithm receives different measurements information in quite different time stamps. In this thesis all the data can be synchronized for the previous IMM approach estimation. In such a way the OOSM problem can just be ignored. However in the real world situation due to different processing time for different features, it is not suitable to have this assumption. Even assuming that all the data are arriving within one time interval, there is also the measurement's time discretization problem, which means that the delayed measurements for the time k may be associated with the time $k - 1$. In order to make the accurate target state estimation, the OOSM problem must be solved. Here with the $1 - \text{step} - \text{lag}$ OOSM estimation algorithm from Bar-Shalom [6], the target state estimation results can be compared from the following figures.

The results of the target's first 15 states are shown to demonstrate the estimation performance with and without the OOSM solution. From Fig.8.12 it is clear that with the $1 - \text{step} - \text{lag}$ OOSM solution, the target's velocity and position estimation performance is improved. The target's position estimation with the $1 - \text{step} - \text{lag}$ OOSM solution is closer to the target's measurement position, while the estimation without OOSM is less accurate. The RMS errors comparison of the estimation results with and without OOSM can be seen in Fig.8.13. In Fig.8.13 more target states are selected to analysis the estimation performance with and without the OOSM solution and the first 15 ones correspond to the states in Fig.8.12. From the figures the RMS errors from the estimation with the OOSM solution are smaller than those without the OOSM solution.

From the figures (Fig.8.12 and Fig.8.14) sometimes there are the OOSM and sometimes there are not. If the delayed measurements at the time k arrive, the measurements which should be associated with the time k may be associated with the time $k - 1$ due to delayed measurements' time association problem. In such a way there are the $1 - \text{step} - \text{lag}$ Out-Of-Sequence Measurements inputs to the Extended Kalman Filter for the time k . When the delayed measurements for the time k arrive with the time stamp $k - 1$, after the state of the target has been already updated to the time k , one faces the problem of updating the current state at the time k with the "older" measurement from the time $k - 1$. The delayed measurements' time association problem is due to

8.12.1:

8.12.2:

Figure 8.12: Target dynamic information estimation results with and without the OOSM solution. One dimensional data along the x axis of the target's 3D world coordinate position and velocity are shown here. The estimation results with the OOSM solution are relatively more accurate than those without the OOSM solution. In this figures the target position measurements are the ground truth values.

8.13.1:

8.13.2:

Figure 8.13: Multiple repetitions of the experiments are performed and the RMS errors are compared. One dimensional data along the x axis of the target's 3D world coordinate position and velocity are shown here. Here the RMS errors from the estimation results with the OOSM solution are relatively smaller.

their discretized process, which is in detail explained next.

8.14.1:

8.14.2:

Figure 8.14: Target dynamic information estimation results with and without the OOSM solution when the target is moving with a direction turn. One dimensional data along the x axis of the target's 3D position and velocity are shown here. In such a situation, the estimation results with the OOSM solution are still quite accurate. In these figures the target position measurements are the ground truth values.

8.15.1:

8.15.2:

Figure 8.15: Multiple repetitions of the experiments are performed and the RMS errors are compared. One dimensional data along the x axis of the target's 3D position and velocity are shown here. Here the RMS errors from the estimation results with the OOSM solution are relatively smaller.

In general, the delayed measurement's time κ doesn't need to correspond exactly to the times $t_{k-1}, t_{k-2}, \dots, t_{k-d}$, and κ might fall within the interval around t_{k-i} , $i \in [1, d]$. If κ

falls inside the interval $[t_k - \frac{\Delta}{2}, t_k + \frac{\Delta}{2})$ centered at t_k , it is approximated that $\kappa = t_k$. $\Delta = t_k - t_{k-1}$. Sometimes the delayed measurement for the time k falls into the current time interval range of t_k , which is then the correct measurement and not considered as OOSM. However sometimes the delayed measurement for the time k falls into the previous time interval range of t_{k-1} , in this case the delayed measurement from the time $k - 1$ needs to be processed, which requires the *1-step-lag* OOSM solution. From the figures the target's dynamic estimates without the OOSM solution are relatively inaccurate and there are larger errors. In this thesis with the *1-step-lag* OOSM solution, the estimates from Extended Kalman Filter are closer to the real measurements, which reflects that the OOSM solution makes uses of the Out-Of-the-Sequence Measurements information and improves the estimation accuracy.

8.3 Experimental Results in the Outdoor Unstructured Environment

In this section, extensive experimental results in the outdoor unstructured environment are shown to analyze the tracking system performances under different situations. Here only the final tracking results are shown, while the step by step image processing and sensor fusion results are omitted.

In the outdoor experiments, the target's position and velocity are estimated in the *3D* world coordinate. In order to show the tracking results, the same method used in the indoor situation is applied here. The target's *3D* position in the world coordinate is projected into the *2D* image through the pinhole camera projection model.

The target tracking system can be used to track any object moving in front of the vehicle. In all the following outdoor experiments the moving objects are people, because tracking moving people is easier for the implementation of the experiments. Also the people's shape is irregular, and it makes the tracking more difficult than the tracking of regular shape objects like cars. The moving people can also be considered as the textured objects which have only the uniform motion properties but quite different color features within their bodies. So if the moving people can be accurately tracked, then

the tracking system is robust to track any other normal moving object.

Since the outdoor environments are very cluttered and complex and the visual features between two consecutive images change greatly, the following experiments are done off-line.

In this experiment, a scene with two people crossing each other is used to analyze the system performance in the occluded target tracking situation. The 3D tracking is

Figure 8.16: Two persons occluded each other and the segmented optical flow vectors fields for this image sequence.

performed and the regions closer to the camera have larger image segments. Here the template size is scaled according to the changes of the visual depth.

However, if the dynamic model of the tracked object is not considered, there is a chance that the algorithm tracks the wrong objects as presented in Fig.(8.17). From Fig.(8.17), first the person on the right is tracked. Then when these two people meet and occlude each other, the tracker is lost. The person moving from left to right is tracked. This may happen if the algorithm uses only the motion fields for the target identification, because at the occlusion point both persons have similar motion regions in the image sequence (Fig.(8.16.5))

In this thesis, the above problem is solved by using a suitable dynamic model for target

Figure 8.17: The tracked target is missed when the two persons cross each other.

tracking, which can estimate and predict the expected direction of the movement of the target. As explained earlier the IMM based multiple linear dynamic models algorithm is used to predict the new possible position of the target in the image sequence. The predicted position of the target with its current position is used to identify the direction of movement (Fig.(8.18)). With the predicted direction the above mentioned problem is avoided as shown in Fig.(8.19).

Figure 8.18: Tracking with a dynamic model direction prediction to solve the occlusion problem.

However there are also several limitations in this occlusion solution by using the motion direction prediction. After the two persons meet and occlude each other, both of them turn their directions and move back to their original places. Or if the two persons are moving with the same motion directions, then one person is moving a little fast and passing the other one. In such situations the motion direction predications may fail to indicate the right direction for the tracked person. From the previous part, the IMM approach can accurately track the person when he is moving with a direction turn, which

Figure 8.19: Tracking performance with the modification for the occlusion situation.

Figure 8.20: Two persons are moving with a moving camera in a cluttered environment. Here when these two persons are moving across each other, the tracker is still good. The person from left to right is tracked through the whole image sequence. There is no missed tracking problem at the occlusion region. The tracking system can accurately track the person when there are several similar motion regions in the image. The occlusion problem is solved using the target's estimated motion direction in the image sequence.

is only due to the reason of using the correct measurements for this person after the direction turn point. If the target tracking is not correct right after the motion direction turn, the system will then track the wrong person continually using the measurements from the wrong target. In these cases the occlusion problem may be solved alternatively using the robust template matching and updating developed in this thesis.

Next several image sequences are used to test the tracking system performance under different situations: the person is moving close to the cameras (Fig.(8.22) and Fig.(8.23));

the cameras are rotating from left to right (Fig.(8.21)); the tracked object is moving with the self rotational motion (Fig.(8.22)); the person and the background's visual features are changing (different lighting conditions) (Fig.(8.23)); two persons are moving together and occluding each other (Fig.(8.24)) and three persons are moving in the cluttered outdoor environment (Fig.(8.25)).

Figure 8.21: The target 3D tracking results when a person is moving in the cluttered environment. Here the camera is rotating from left to right. This is considered as the cameras' both rotational and translational motions situation. The person is correctly tracked with the proposed system.

From this image sequence (Fig.(8.24)), the two persons are moving with the same motion directions. The previous motion direction predication method can not solve the occlusion problem here. The occluded target tracking problem is solved using the robust template matching and updating algorithm developed in Chapter 6.

Several image sequences (Fig.(8.26) and Fig.(8.27)) are shown next to demonstrate the experimental performance of the tracker automatically initialization, termination and multiple persons tracking.

The proposed target tracking scheme is based on the image motion vectors. Here if a new person is entering the image scene, the tracker can be automatically initialized for this person. When the person is leaving the image scene, the tracker can also be

Figure 8.22: The target 3D tracking results when a person is moving in the outdoor unstructured and cluttered environment. The person is self-rotationally moving and the cameras are also moving. The person's visual features are changing through the image sequence. Here the person is correctly tracked with its visual features' changes.

automatically terminated. The tracking results are good as seen in Fig.(8.26).

In Fig.(8.26) and Fig.(8.27), the tracking system is tested under the situations when several different persons enter and leave the image scene. In such a situation, the tracking system can track different people at the same time in the same environment. If several persons are moving in the environment and a new person is entering the image scene, the target tracking strategy is set so that the pervious tracked person's appearance is memorized as the image template, and then a new tracker different from the previous tracked one is initialized for the newly entered motion region in the images. From the experimental results the system can accurately initialize or terminate a tracker for each person. This reflects that the tracking system tracks different objects at the same time independent of their visual features. In Fig.(8.27) several people have the same color properties. Here the tracked person's 3D velocity and position in the world coordinate are estimated by Extended Kalman Filter. With the estimated person's position and velocity, the tracking can be accurately done and there are no wrong tracking with other persons.

Figure 8.23: The target 3D tracking results when the person is moving in the cluttered outdoor environment. The person is moving from the dark side to the bright side. In such a situation the visual features of the person are changing. The person looks much brighter frame by frame. If the tracking is done without the motion based algorithm developed in this thesis, the tracking results are not good and the tracker is lost (Fig.(8.23.4) and Fig.(8.23.5)). Here the performance is improved with the proposed motion based tracking algorithm as shown in Fig.(8.23.6), Fig.(8.23.7), Fig.(8.23.8), Fig.(8.23.9) and Fig.(8.23.10). The person is accurately tracking with its various visual features.

In the outdoor unstructured environment the human is tracked by autonomous vehicles. Compared with the 2D position estimation result of Fig.(8.7) in the indoor structured environment, Fig.(8.28) shows the person's 3D position estimation results in the world coordinate. Fig.(8.29) shows the person's 3D position estimation results in the world coordinate, when the person is moving with a direction turn. From these figures the person's 3D movement in the world coordinate can still be accurately obtained for autonomous vehicle navigation.

8.4 Experimental Results Discussion

Fig.(8.19), Fig.(8.20), Fig.(8.21), Fig.(8.22), Fig.(8.23), Fig.(8.24), Fig.(8.25), Fig.(8.26) and Fig.(8.27) show the proposed target tracking scheme's performance in different out-

Figure 8.24: Two persons are moving with the moving cameras in a cluttered environment. Here when the first person is fully occluded by the second person, the tracker is lost. With the proposed Bayesian fusion based template matching and updating algorithm in Chapter 6, when the occlusion is finished, the tracking template is updated together with the first template and the right person is still being tracked. In such a situation the tracking algorithm still can give good performance.

door unstructured situations with both the moving cameras and moving objects. Here first, the moving object can be seen as the textured object in an outdoor unstructured and cluttered environment. Also sometimes the visual features of the objects of interest are changing frame by frame. The object identification is accurately done with the fusion based image segmentation and clustering and the robust template matching algorithms. Secondly, as the cameras and objects are moving towards each other, the cameras not only have the pure translational motion but also have the rather complex 3D motions. In this case the target 3D velocity and position in the world coordinates are estimated with the cameras' rotational and z-translational motion, which is based

Figure 8.25: The target's 3D tracking results when three persons are moving in the cluttered outdoor environment. One person is correctly tracked. However when the tracked person leaves the image scene, the tracker is lost. Since the proposed method is based on the motion features tracking, then the algorithm tracks the person with the most similar motion instead. (Fig.(8.25.6) and Fig.(8.25.7))

Figure 8.26: The target tracking results when several different people enter the image scene. The cameras are also moving in an indoor environment. Here the tracker is initialized when one person is entering the image scene and the tracker is terminated when this person is leaving the image scene.

Figure 8.27: The target tracking results when several people are moving in the same environment. The cameras are also moving. As same as Fig.8.26, the tracker can be automatically initialed and terminated for each person. Here the tracking can be accurately done when several people move in the same environment. Even if two persons have quite similar visual properties (Fig.(8.27.8), Fig.(8.27.9) and Fig.(8.27.10)), the tracking can still be accurately done.

on the moving cameras' kinematics model (R from Eq.(7.7) for the rotational motion and T from Eq.(7.8) for the translational motion.). In these cases the target tracking is still quite accurate. Thirdly, in Fig.(8.24) the motion of the cameras looks large while in Fig.(8.25) the cameras look to be moving with a slow velocity. When the cameras' motion looks large, the changes between two consecutive images are relatively larger than those with the cameras' slow motion. In this thesis, all the data processing are done off-line, so even with a large amount of data to be processed the tracking system can well deal with the large motion of the cameras. Fourthly, in Fig.(8.23), since the environmental lighting condition changes greatly, the target's visual features change a lot through the image sequence. The proposed tracking system is designed based on the object's motion properties, which is independent of other visual features like color. Fifthly, the tracking system can accurately initialize and terminate a tracker when a new object enters or leaves the image scene. The previous tracker can also be memorized by its appearance template and then new trackers can be initialized for the newly entering objects when there are already several targets moving in the image scene. From the

8.28.1:

8.28.2:

8.28.3:

Figure 8.28: The person's estimated 3D position in the world coordinate. Fig.(8.28.1) shows the person's projected 2D position in the $X - Y$ plain. Fig.(8.28.2) shows the person's projected 2D position in the $X - Z$ plain.

experimental results, in all these five cases the target tracking is still accurate.

In the proposed target tracking scheme the IMM approach and OOSM solution are used to improve the target tracking performance. Here several issues need to be discussed for

8.29.1:

8.29.2:

8.29.3:

Figure 8.29: The person's estimated 3D position in the world coordinate. Here the person is moving with a direction tune not along a straight line as shown in the Fig.(8.28). Fig.(8.29.1) shows the person's projected 2D position in the $X - Y$ plain. Fig.(8.29.2) shows the person's projected 2D position in the $X - Z$ plain.

the IMM and OOSM.

The Interacting Multiple Models (IMM) approach computers the state estimates that

account for each possible current model using a suitable mixing of the previous model-conditioned estimates depending on the current model. The algorithms are decision free so no maneuver detection decision is needed. From the experiments the performance of the IMM algorithm is not very sensitive to the choice of the transition probabilities. Thus in this thesis the model transition probabilities (Eq.7.22, Eq.8.18 and Eq.8.19) are assumed time invariant and independent of the base state which means a homogeneous Markov chain.

For the general OOSM estimation there are some issues needed to be considered. First the optimal algorithm for updating the state estimate with an (earlier) OOSM requires more than the current (latest) state estimates. If the time of the OOSM is within the last sampling interval, one has to store the last innovation to carry out the optimal retrodiction to the time of the OOSM. This amounts to a (non-standard version) smoothing, which always requires preprocessing the measurements. Secondly, in practical systems, because of the process noise the informational content of a measurement diminishes rapidly when its age-old data are irrelevant. Only if the measurement is not too old, it is relevant enough for updating. So here the approximation of 1 – *step – lag* OOSM problem in this thesis is necessary to make an accurate performance.

In view of the need to store and reprocess measurements to incorporate optimally an OOSM, one option is to carry out the reprocessing in the chronological sequence. To obtain a strict chronological sequence of the measurements is possible, however, only if there is no data association problem (DAP) to solve simultaneously with the estimation problem. If there are the DAP and communication delays, one can store sensors scans and reprocess them in their chronological order after some earlier scans arrive. These scans can be ordered according to their end-times. However, as shown in detail in [214], this still leaves room for OOSM, but the “negative times” to consider are less than a sampling interval if the various sensors have the same sampling rate, i.e., the situation described in Eq.7.35. This motivates the focus of the present work on this problem.

The proposed visual target tracking system is tested under different situations and demonstrates good experimental performance. However there are still several limitations in the proposed visual tracking scheme.

- The autonomous vehicle is assumed to move in the environment with the prior knowledge of its localization. The vehicle's localization information is used to estimate the target's 3D position and velocity in the world coordinate. Since the task in this thesis is to design the visual tracking system for autonomous vehicle navigation, the SLAM (Simultaneously Localization and Map Building) problem is not considered.

Wang et. al. [213] integrate SLAM and target tracking methods and successfully demonstrate SLAM with target tracking at high speeds in the large crowded city environments. So in order to make the vehicle fully autonomously navigate in the environment and track the object, the SLAM problem can be considered at the same time with the target tracking problem.

- The cameras are calibrated before the experimental testing. All the calibration parameters are kept unchanged for the experiments. The cameras are fixed on the moving platform and the cameras normally can not move, zoom in or zoom out. There is also no pan/tilt movement for the cameras. However the view of stereo cameras is limited due to the cameras capability. The object must be seen by both the two cameras. Sometimes when the object is moving too fast, not far enough away from the cameras or moving in the opposite or parallel directions as the cameras, there will be no object seen in the cameras scene. The system requires the object to be seen in the image scene for tracking. If the objects are not there, the tracking algorithm will decide that the object exits the image scene, stop the following image processing and terminate the target tracking.
- The computational time is another important issue to be considered for the tracking system. Currently the experiments are done off-line. A moving platform is just designed with all the sensors and embedded pc available to simulate an autonomous vehicle. Building a real vehicle with fully autonomous functions is really difficult. But from the off-line experiments, the results are good under different situations. And the theories of this thesis are also sound. So after new hardware are purchased and a whole autonomous vehicle is build, the proposed algorithms can be successfully implemented to get the satisfactory real-time performance under the complex situations. Another reason is that the real-time part of the experiments

is only that after the target has been tracked, the target's dynamic information is just sent to the vehicle's motion controller and it will generate commands to control the vehicle's movement to track or follow the target. The vehicle's kinematic control is really out of the scope of this thesis. This thesis is to design a computer vision and sensor data fusion based target tracking system, and for this part it has successfully fulfill the task.

In fact the computational requirement for a normal target tracking system is always one of the major limitations. The system always wants to have a fast processing algorithm. But with a fast algorithm the performance will probably be less accurate. There is always a trade-off. It is just assumed that in the near future the computer hardware will have a tremendous development and the computation will not be a real problem any more.

Currently, the system is limited to the low speed testing of both the moving vehicle and objects. In the experimental testing the off-line data processing is done with Matlab. Matlab programming is quite time consuming. From the images both the moving platform and the objects are not moving very fast. In such a slow movement situation, the computational time from Matlab is already very big. It takes about 1.5 minutes using Matlab to process 1 second of video images. With much more powerful hardware, the system can be implemented into a very fast moving situation.

- In the proposed target tracking system, there are the accumulated odometry errors, and T_C^W and R_C^W will become increasingly inaccurate as the time goes by. So here there should be a mechanism to reset these accumulated odometry errors. In the proposed target tracking system, the experimental testing is limited to a not very long time performance. After some time, the parameters should be reset to correct the accumulated odometry errors.
- From the experimental results presented in this Chapter, the tracking system can accurately and automatically initialize a tracker when an object enters the image scene. However if there are several objects moving in the same environment at the same time, the object of interest should be first identified by human. Then

the moving platform tracks it wherever it moves or stops. In such a situation it requires the tracking system to have the prior knowledge of which target is of interest. This knowledge is not the special visual features of the objects but only the information of which object to be tracked. In the real world applications, this prior knowledge should also be provided by human beforehand. For example, in the intelligent transportation system, the system do not have the intelligence to decide which vehicle to track or which obstacle to avoid if there are a lot of vehicles or obstacles encountered in the environment. For the video surveillance system, the system can not yet decide which person is dangerous and needs to be tracked. All these intelligences must be provided by human. Currently, it is necessary to initialize the tracking manually in those difficult situations. In the future the high level image processing algorithms like artificial intelligence or human reasoning can be developed for the visual target tracking system. The human motion analysis can be done for the system to recognize the human's actions and decide which person is of interest and should be tracked. This is considered as the future work.

- In the proposed target tracking system, there are the accumulated odometry errors, and T_C^W and R_C^W will become increasingly inaccurate as the time goes by. So there should be a mechanism to reset these accumulated odometry errors. In the proposed target tracking system, the experimental testing is limited to a not very long time performance. After some time, the parameters should be reset to correct the accumulated odometry errors.
- From the experimental results the tracking system performs quite well in different situations. In fact sometimes the tracking system may fail to correctly track the object. In the 2D images the rectangle can not accurately fit the object's region in the images. These failures are due to various reasons.

The tracking scheme developed in this thesis is a bit complex. Several different image processing algorithms are developed step by step to identify and track the objects. There are also some parameters in these algorithms. If one image processing step or one parameter is wrong, then the whole tracking system can not correctly track the object. In fact currently the experimental testings are done off-line, so all the processing steps can be carried out separately. The parameters can

be adjusted during the processing under different situations. But in the real-world situation, the whole image processing and fusion steps are conducted continually and automatically. Human is not supposed to change anything once the tracking is started.

The performance of the whole tracking system is limited due to its complexities and the parameters. Because the visual target tracking for autonomous vehicle navigation is a very difficult research problem, a simple method definitely can not give accurate and robust results. More complex and adaptive methods are required to be developed. The trade off here is that more complexity means more chances to get wrong and less complexity means less accuracy and robustness. In the near future the systems must be modified and more thoroughly tested to solve the tracking failure problems and improve its robustness under complex and difficult situations.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

At present vision based target tracking has many applications for autonomous vehicle navigation, which includes the intelligent transportation systems, autonomous vehicle path planing and obstacle avoidance systems, mobile robots soccer, underwater investigation systems and so on. The work described in this thesis proposes a visual target tracking scheme for these potential applications. The entire scheme is designed based on the fusion of data from the vehicle's onboard CCD cameras and inertial motion sensors to estimate and track the target's $3D$ dynamics in the world coordinate.

In Chapter 2, literature reviews and research challenges on this topic are summarized. In Chapter 3, a data fusion based visual target tracking scheme is proposed as a novel and better solution for the vision based autonomous vehicle navigation. In the followed chapters, this proposed scheme is described and the remaining research problems are solved. Good experimental performance is also demonstrated in each Chapter.

The target's $3D$ visual features (optical flow vectors and stereo disparities) are first extracted from the images for target tracking. In this thesis the optical flow motion vectors field is used as the major source for tracking a moving object. In Chapter 4 three commonly used optical flow estimation algorithms (Horn and Schunck's method, Lucas and Kanade's method and the structure tensors and parametric model method)

are studied and tested. Based on the experimental testing, Horn and Schunck's classic gradient descent based method is selected for the proposed visual target tracking scheme. Horn and Schunck's method is easy to implement, fast in computation. The optical flow vectors field from Horn and Schunck's method is 100% dense, which is also good to reflect the target and background motions for image segmentation and clustering. At the same time, the target's 3D visual depth information is estimated from stereo images by using the commonly Sum of Absolute RGB difference Disparity (SAD) estimation method.

Next, in order to accurately identify the target's useful 2D features from the images, two novel fusion based image processing algorithms are developed in this thesis.

- In Chapter 5 the target's color features, spatial locations and motion vectors are fused for the region of interest extraction from the image sequence with a changing background. Different visual features are combined to make a robust image segmentation and clustering. Two different image segmentation methods (K-means based and unsupervised Bayesian on-line learning based) are presented in this Chapter. The K-means method is easy to implement and fast in computational time, but it is less accurate and can not give satisfactory results which meet the temporal consistency constrain requirements for video images segmentation. In the unsupervised Bayesian on-line learning based algorithm, the image pixels clustering is done in a probability based framework. Each visual feature's cluster is assigned with a Gaussian PDF. Then the unsupervised Bayesian on-line learning method is developed to learn and update the image cluster's PDF parameters. The Bayesian estimation uses all the previous available information to compute the desired density $p(x|D)$ for the next image clustering. The linear complexity of the Bayesian parameters estimation and good experimental results make it a good choice for the object-based video images segmentation, which is suitable for the proposed visual target tracking scheme.

Here one big challenge in the image clustering is to determine the cluster number. In Chapter 5 the cluster validity measures based on the K-means image clustering method are introduced. This method overcomes the limitation of having to indicate

the number of clusters by incorporating a validity measure based on the intra-cluster and inter-cluster distance measures. Based on the cluster validity measures, the cluster number can be normally set to be around 3 – 5 for natural image sequences segmentation.

- After the image sequence is successfully segmented and clustered, Chapter 5 presents the template matching methods to localize the region of interest in the clustered images. In Chapter 5, first, the normally used Sum of Absolute Difference (correlation) based template matching method is implemented to match the optical flow vectors template to the image's clustered optical flow vectors field. Next, a more robust algorithm is proposed to improve the matching performance. The proposed method is developed based on fusion of different matching methods through the Bayesian probability theory. SSE template matching method is selected as the basic method. The GMMs is then combined with SSE to solve the problems of the template appearance changes or the object occlusion. A robust template updating method is also proposed based on the Bayesian fusion matching. Here the template is updated together with both the SSE and GMMs matching PDF parameters.

Through experimental results, the fusion of two different template matching methods solves the template matching problems such as the template drifting, shape rotation, appearance changes or occlusions. Since the algorithm can make accurate template matching and target tracking when the cameras are also moving (Fig.(6.9)), it is suitable for the proposed visual target tracking scheme [43].

After the target's 2D visual features are identified from the image sequence, the target's 3D dynamic information in the world coordinate is estimated based on the sensor data fusion algorithm developed in this thesis. In Chapter 6 the 2D target visual information is fused with the target's 3D disparity depth information through the pinhole camera projection model. Next, the extracted target 3D visual features are fused with the camera motion parameters for the target's 3D velocity and position estimation in the world coordinate.

In Chapter 7 different visual features and sensor data are combined through the Extended Kalman Filter measurement equation for the target's 3D dynamics tracking. In Chapter

7 two estimation methods are developed to improve the Extended Kalman Filtering performance.

- Since normal moving objects have complex or unknown motion properties, several simple and basic linear dynamic models are fused in Extended Kalman Filter through the IMM algorithm, which provides an accurate approximate model for the target's complex dynamics. Through IMM, the performance of the target dynamic state estimation is improved.
- Because the proposed system is based on sensor data fusion, there is the OOSM (delayed measurement) target state estimation problem. In this thesis the "1-step-lag" OOSM estimation solution from Bar-Shalom is developed for Extended Kalman Filtering. With the OOSM solution, the target tracking performance becomes more robust.

With the object's 3D velocity and position information, the visual target tracking can be finally achieved for autonomous vehicle navigation.

Through the final target tracking experiments in Chapter 8, the proposed tracking scheme demonstrates satisfactory performances under different situations. The system can well deal with the situations such as: moving cameras and moving objects' tracking; the tracking with the object's rotational motion; the tracking with the cameras' rotational or z-translational motion; the target occlusion tracking; moving targets with highly nonlinear or unpredictable dynamics; the tracking in both the structured and unstructured environment; the environmental condition changes; the target tracker's automatically initialization, termination or multiple targets tracking and so on.

Compared with other methods [43] [88] [145], this tracking system can accurately detect and track the target's 3D dynamics in the MCMO state. In conclusion the proposed visual tracking scheme has the following advantages over the other methods.

1. The tracking system is designed only based on the vehicle's onboard sensors without any additional ones in the environment to detect and track the object.
2. The object is detected based on its motion properties rather than any special visual

properties, which makes the tracking system solve the problems of the unknown or textured object tracking or the multiple objects tracking (automatically tracker initialization and termination).

3. The image processing algorithms in this thesis are developed based on data fusion. In the image segmentation section the image color, motion vectors and spatial location are fused together for robust image clustering. In the template matching section, different template matching algorithms are fused together through the Bayesian probability theory for robust template matching and updating. Through data fusion the image segmentation and template matching experimental performances are greatly improved.
4. In the image segmentation and clustering section, the target is segmented out from the 2D image frame with the unsupervised Bayesian on-line learning. The Bayesian learning updates the image clustering functions frame by frame, which adapts the image classifier with the visual features changes in the MCMO state. In such a way the object can be clustered into the same image cluster through the whole video images sequence, which makes the target identification an easy task. This is also suitable to meet the temporal consistency constrains requirement for the video images segmentation.
5. In the image visual features localization section, the Bayesian probability theory is used to combine different template matching methods. The Sum of the Squared Errors (SSE) template matching is fused with the Gaussian Mixture Models (GMMs) to conduct the robust image template matching and updating. In this way the template's shape rotation, appearance changes and occlusion matching problems can be accurately solved. Through the experimental results, the combination of different methods has shown good performance under different template matching situations.
6. Compared with the previous visual target tracking systems designed for autonomous vehicles, the target's 3D dynamics in the world coordinate rather than its relative dynamic information to the cameras is estimated for tracking. The 3D dynamics in the world coordinate is more accurately estimated than the relative information

because the vehicle's self movement is considered in the $3D$ world coordinate. Also the object's $3D$ dynamics can be more generally used by other applications in the same environment.

7. The kinematic model of the autonomous vehicle is included in the tracking system. The kinematic model provides the movement information of the cameras. The cameras' motion parameters are fused with the target visual features, which are then used to estimate the target $3D$ dynamics in the world coordinate. The inclusion of the vehicle's kinematic model into the tracking system makes the target's $3D$ dynamics estimation more accurate and also achieves the target tracking with the cameras' complex motion.
8. Fusion of several simple linear dynamic models through the IMM approach makes the target dynamics tracking more robust and accurate than only using the single linear models. The tracking system can well deal with the situation when the target is with complex, unknown or highly nonlinear dynamic properties.
9. The proposed algorithm is designed based on sensor data fusion. However there is generally the Out-Of-Sequence Measurements problem. In this thesis the $1 - \text{step} - \text{lag}$ OOSM solution is developed with the Extended Kalman Filter to solve the delayed measurements problem and improve the system's performance.

In conclusion it can be stated that the proposed visual target tracking scheme is suitable for the target identification and tracking for autonomous vehicle navigation.

In this thesis the proposed system has shown quite satisfactory off-line results through experimental testing under different situations. However there are also limitations encountered in this tracking system.

- Due to the computational cost issue, the system is limited to the low speed testing of both the moving vehicle and objects. The computational time is rather large with Matlab programming for the off-line testing.
- The SLAM problem is not considered in this thesis. The knowledge of the vehicle's location in the environment is assumed to be known beforehand.

- The relative movements between the vehicle and target are restricted. If the cameras can not see the object, the tracking scheme will terminate the tracker for this object.
- The objects of interest must be first initialized by human if there are several objects together moving in the image scene. The tracking system can automatically initialize a tracker only if one new object is moving or entering the image scene.
- The tracking system sometimes may fail. The reasons are mainly due to the image conditions, the robustness of the image processing algorithms and the accumulated odometry errors. For example, if several similar objects are together moving in the image scene and they cross each other continuously, in such objects highly occluded situation, the tracking will probably fail.

When the visual target tracking system is implemented for the real autonomous vehicle experiments, the real-time performance is the first thing to be considered. Today with powerful processing computer systems readily available, the real-time target tracking should not be a problem. The accuracy of the tracking system in very complex experimental environments is another important issue to be considered. All the work to be done in the future is discussed in the next section.

9.2 Future Work

As for the future work, each part of this tracking system should be modified to improve the tracking system performance.

- A “smarter” and faster tracker initializing and updating algorithm can be developed to make the target tracking more accurate and robust. High level image processing and pattern recognition algorithms can be developed for the visual tracking system. The tracking system will have the ability to recognize and analyze the human’s actions. In such a way if there are several persons moving in the environment, the tracking system can automatically initialize the tracker based on the human motion analysis.

- In order to obtain the best IMM based Extended Kalman Filtering, the IMM algorithm has to be properly improved to meet the following requirements: (1) design of the target motion models for all modes of movement considering both the quality and complexity of the model; (2) selection of the model parameters, such as the noise level; (3) determination of the parameters of the underlying Markov chain, that is, the transition probabilities.
- The algorithm for incorporating the OOSM into an IMM estimator from [7] can be introduced into the proposed visual target tracking scheme. Combining the OOSM solution with the IMM approach makes the system rather more complex, but the whole system performance can be improved to some extent.
- The accumulated odometry errors should be solved by designing some parameters resetting strategies. In such a way, the target tracking will not become increasingly inaccurate as the time goes by.
- Here the tracking system is designed based on data fusion of optical features from the CCD cameras and the vehicle's inertial motion sensors data. In the future CCD cameras fused with other sensors such as RADAR or LASER can be introduced into this tracking scheme. CCD cameras can be used to derive the target's $2D$ visual information. The RADAR or LASER can help to obtain the target's $3D$ depth information.
- The tracking system developed in this thesis is for the applications of autonomous vehicle navigation. The tracked object's $3D$ velocity and position information is used for goal seeking, obstacle avoidance or target following applications. So finally the system should be implemented in the real-time autonomous vehicle experiments. In the future the proposed algorithm can be further tested in the land autonomous vehicle and Autonomous Underwater Vehicle (AUV) available in Nanyang Technological University. They are the Generic Outdoor Mobile Explorer (GenOME) (Fig.(9.1)) and the Nanyang Technology University-Underwater Autonomous Vehicle (NTU-UAV) (Fig.(9.2)).
- This visual tracking system is designed mainly based on the optical flow vectors and has the ability to deal with different tracking conditions. However if to track the

9.1.1:

9.1.2:

Figure 9.1: Generic Outdoor Mobile Explorer (GenOME) in NTU

9.2.1:

9.2.2:

Figure 9.2: NTU-UAV

objects in the real-world conditions with 7 days and 24 hours accuracy, the optical flow based tracking system has to overcome a lot of other practical problems. These problems are also considered as the future work.

1. Image Noise. Image noise discontinuities in the image intensity levels between frame captures can cause errors in the motion field estimation.
2. Occlusion Problem. This is caused by the partial or complete overlap between multiple objects in a scene. In this thesis the occlusion problem is solved in some extent based on the target's motion dynamic models and also the

robust template matching and updating methods. However there are still some difficult and complex occluded object tracking situations to be studied and solved in the future work.

3. Problem of aperture. This problem occurs when considering only the local motion of an object. The aperture problem is such that, only the component of the real velocity, normal to an edge can be detected and estimated with any form of accuracy. The aperture problem affects the optical flow estimation, which needs to be studied and solved in the future work .
4. Rotation Problem. In this research the object moving with the simple rotation velocity is tested. However if the object has a 3D and six degrees of freedom rotation velocity, the optical flow vectors field cannot accurately reflect its real movement in the images. Thus the motion based tracking system may give inaccurate results under this situation. This should also be carefully studied and solved in the future.

Author's Publications

1. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Sensor Fusion Based Vision Target Tracking For Autonomous Vehicles.", Accepted for International Journal of Robotics and Automation.
2. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Visual Information Fusion for Object Based Video Segmentation Using Unsupervised Bayesian Online Learning.", Accepted for IEE Proc. Vision, Image & Signal Processing.
3. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Visual Data Fusion based Target Tracking for Autonomous Vehicles with the OOSM Solution.", Revised for Robotics and Autonomous Systems (minor changes).
4. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Visual Motion Field based 3D Target Tracking using Interacting Multiple Dynamics Models.", Submitted to International Journal of Computer Vision and Image Understanding (Revised).
5. Zhen Jia, Laurent Amsaleg and Patrick Gros, "Ultra-fast content-based image retrieval from a very large image database.", Submitted to Pattern Recognition.
6. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Template Matching for Target Tracking from the Bayesian Probability Fusion of Two Different Matching Algorithms.", Submitted to Image and Vision Computing.
7. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Vision based Target Tracking for Autonomous Vehicles Navigation: A Survey.", Submitted to Autonomous Robots.

8. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Vision based Target Tracking for Autonomous Vehicles Navigation: A Brief Survey.", Accepted for the 2006 IEEE International Conference on Systems, Man, and Cybernetics, Oct.8-Oct.11, 2006-The Grand Hotel, Taipei, Taiwan.
9. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Visual Target Tracking for Autonomous Vehicles with the Optimal OOSM Solution.", In the Proceedings of The 8th International IEEE Conference on Intelligent Transportation Systems-ITSC 2005, Vienna, Austria, September 2005.
10. Zhen Jia, Arjuna Balasuriya, "Target Tracking With Bayesian Fusion Based Template Matching.", In the Proceedings of IEEE International Conference on Image Processing (ICIP) 2005, Genoa, Italy, September 11-14, 2005.
11. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Sensor Fusion based 3D Target Visual Tracking for Autonomous Vehicles with IMM.", In the Proceedings of 2005 IEEE International Conference on Robotics and Automation (ICRA), pp.1841-1846, April 18-22, 2005. Palau de Congresses de Catalunya, Barcelona, Spain.
12. Zhen Jia, Arjuna Balasuriya, "Motion Based Image Segmentation with Unsupervised Bayesian Learning.", In the Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision and the IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), volume 2, pp.2-7, Jan 5-7 2005, Breckenridge Colorado, USA.
13. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Visual 3D Target Tracking for Autonomous Vehicles.", In the Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS), pp.820-825, December 1-3, 2004 Singapore.
14. Zhen Jia, Arjuna Balasuriya, "Bayesian Learning for Object Based Image Segmentation.", In the Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS), pp.762-767, December 1-3, 2004 Singapore.
15. Zhen Jia, Arjuna Balasuriya and Subhash Challa, "Motion based 3D Target Tracking with Interacting Multiple Linear Dynamic Models.", In the Proceedings of The 15th British Machine Vision Conference (BMVC 2004), London, UK.

16. Zhen Jia, Arjuna Balasuriya and Subhash Challa, “Visual Information and Camera Motion Fusion for 3D Target Tracking.”, In the Proceedings of The Eighth International Conference on Control, Automation, Robotics and Vision, ICARCV 2004, pp.2296-2301, Kunming, P.R.China.
17. Zhen Jia, Arjuna Balasuriya and Subhash Challa, “Vision Data Fusion for Target Tracking.”, In the Proceedings of The 6th International Conference on Information Fusion, volume 1, pp.269-276, 8-11 July 2003, Cairns, Queensland, Australia.
18. Arjuna Balasuriya, Zhen Jia and Subhash Challa, “Data Fusion for Target Tracking by Unmanned Underwater Vehicles.”, In the Proceedings of UDT Asia 2003, 12-14 November, Singapore.
19. Zuo-Jun Liu, Wei-Hong Qi, Zhen Jia, Peng Huang, “System Integration Control of HVAC in Intelligent Building.”, In the Proceedings of the Third IEEE International Conference on Machine Learning and Cybernetics, pp.1125-1128, Shanghai, 26-29 August 2004.

Bibliography

- [1] J.K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Non-rigid motion analysis: Articulated & elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, 1998.
- [2] Y. Altunbasak, P.E. Eren, and A.M. Tekalp. Region based parametric motion segmentation using color information. *Journal of Graphical Models and Image Processing*, 60(1):13–23, January 1998.
- [3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.
- [4] A. Balasuriya and T. Ura. Underwater robots for cable following. In *Proceedings. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 1811–1816, Wailea, Hawaii, USA, October 2001.
- [5] B.A.A.P. Balasuriya. *Computer Vision for Autonomous Underwater Vehicle Navigation*. Ph.D. Thesis, Department of Naval Architecture and Ocean Engineering, Faculty of Engineering, University of Tokyo, Tokyo, Japan, 1998.
- [6] Y. Bar-Shalom. Update with out-of-sequence measurements in tracking: Exact solution. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):769–777, July 2002.
- [7] Y. Bar-Shalom and H.M. Chen. Imm estimator with out-of-sequence measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):90–98, January 2005.

- [8] Y. Bar-Shalom, H.M. Chen, and M. Mallick. One-step solution for the multi-step out-of-sequence-measurement problem in tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 40(1):27–37, January 2004.
- [9] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley, New York, USA, 2001.
- [10] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [11] A.M. Baumberg and D.C. Hogg. “*Learning Deformable Models for Tracking the Human Body*” in *Motion-Based Recognition*, M. Shah and R. Jain Editions. Kluwer, Norwell, MA, USA, 1996.
- [12] J.E. Bennett, A. Racine-Poon, and J.C. Wakefield. “*MCMC for Nonlinear Hierarchical Models*” in *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson and D.J. Spiegelhalter Editions. Chapman and Hall, London, U.K., 1996.
- [13] L. Bergen and F. Meyer. A novel approach to depth ordering in monocular image sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 536–541, Hilton Head Island, South Carolina, USA, June 2000.
- [14] A. Betser, P. Vela, and A. Tannenbaum. Automatic tracking of flying vehicles using geodesic snakes and kalman filtering. In *Proceedings of 43rd IEEE Conference on Decision and Control*, volume 2, pages 1649–1654, Atlantis, Paradise Island, Bahamas, December 2004.
- [15] J.C. Bezdek and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics, Part B Cybernetics*, 28(3):301–315, June 1998.
- [16] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 568–574, San Juan, Puerto Rico, June 1997.

-
- [17] A.J. Briggs, D. Scharstein, D. Brazianus, C. Dima, and P. Wall. Mobile robot navigation using self-similar landmarks. In *Proceedings of ICRA '00. IEEE International Conference on Robotics and Automation*, volume 2, pages 1428–1434, San Francisco, CA, USA, April 2000.
- [18] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, August 2003.
- [19] D. Burschka and G. Hager. Scene classification from dense disparity maps in indoor environments. In *Proceedings of 16th International Conference on Pattern Recognition*, volume 3, pages 708–712, Quebec City, Canada, August 2002.
- [20] D. Burschka and G. Hager. Principles and practice of real-time visual tracking for navigation and mapping. In *Proceedings of International Workshop on Robot Sensing. ROSE 2004*, pages 1–8, May 2004.
- [21] R. Canals, A. Roussel, J.-L. Famechon, and S. Treuillet. A biprocessor-oriented vision-based target tracking system. *IEEE Transactions on Industrial Electronics*, 49(2):500–506, April 2004.
- [22] S. Challa, R. J. Evans, and X.Z. Wang. A bayesian solution and its approximations to out-of-sequence measurement problem. *Journal of Information Fusion*, 4:185–199, September 2003.
- [23] J.Y. Chang, W.F. Hu, M.H. Cheng, and B.S. Chang. Digital image translational and rotational motion stabilization using optical flow technique. *IEEE Transactions on Consumer Electronics*, 48(1):108–115, February 2002.
- [24] M.M. Chang, A.M. Teklap, and M.I. Sezan. Simultaneous motion estimation and segmentation. *IEEE Transactions on Image Processing*, 6(9):1326–1333, September 1997.
- [25] J.Q. Chen, T.N. Pappas, A. Mojsilovic, and B.E. Rogowitz. Adaptive perceptual color-texture image segmentation. *IEEE Transactions on Image Processing*, 14:1524–1536, October 2005.

- [26] J.C. Cheng and J.M.F. Moura. Capture and representation of human walking in live video sequence. *IEEE Transactions on Multimedia*, 1:144–156, June 1999.
- [27] A. Clerentin, L. Delahoche, E. Brassart, and C. Cauchois. Mobile robot localization based on multi target tracking. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 1, pages 13–18, Washington, DC, USA, May 2002.
- [28] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research: Part C*, 6:271–288, 1998.
- [29] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89:1456–1477, October 2001.
- [30] D. Coombs and C. Brown. Real-time smooth pursuit tracking for a moving binocular robot. In *Proceedings of 1992 IEEE Conference on Computer Vision and Pattern Recognition*, pages 23–28, Urbana IL., USA, June 1992.
- [31] G. Csurka and P. Bouthemy. Direct identification of moving objects and background from 2d motion models. In *Proceedings of International Conference of Computer Vision*, volume 1, pages 566–571, Kerkyra, Corfu, Greece, September 1999.
- [32] R. Cucchiara, A. Prati, and R. Vezzani. Object segmentation in videos from moving camera with mrfs on color and motion features. In *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 405–410, Madison, Wisconsin, USA, June 2003.
- [33] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):781–796, August 2000.
- [34] T. Dang, C. Hoffmann, and C. Stiller. Fusing optical flow and stereo disparity for object tracking. In *Proceedings of the 5th IEEE International Conference On Intelligent Transportation Systems*, pages 112–117, Singapore, September 2002.

- [35] N.X. Dao, B.-J. You, S.-R. Oh, and M. Hwangbo. Visual self-localization for indoor mobile robots using natural lines. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1252–1257, Las Vegas, Nevada, USA, October 2003.
- [36] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 716–721, Kerkyra, Greece, 1999.
- [37] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81(3):328–357, 2001.
- [38] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 558–594, Ft. Collins, CO, USA, 1999.
- [39] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society B*, 39(1):1–38, 1977.
- [40] H.W. Deng and D. A. Clausi. Unsupervised segmentation of synthetic aperture radar sea ice imagery using a novel markov random field model. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):528–538, March 2005.
- [41] S. Derrouich, K. Izumida, K. Murao, and K. Shiiya. The use of cnn models and vertical rectification for a direct trigonometric recovery of 3d scene geometry from a stream of images. In *Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1600–1604, Asilomar, CA., USA, November 2003.
- [42] S. Derrouich, K. Izumida, and K. Shiiya. A combination of monocular ccd camera and inertial-sensor for range estimation. In *Proceedings of IEEE 2002 28th Annual Conference of the IECON [Industrial Electronics Society]*, volume 3, pages 2191–2196, Sevilla, Spain, November 2002.

- [43] G.N. DeSouza and A.C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, February 2002.
- [44] J. Ding, H. Kondou, H. Kimura, Y. Hada, and K. Takase. Robust tracking for camera control on an irregular terrain vehicle. In *Proceedings of the 41st SICE Annual Conference on SICE 2002*, volume 2, pages 1191–1196, Osaka, Japan, August 2002.
- [45] M. Doi, M. Nakakita, Y. Aoki, and S. Hashimoto. Real-time vision system for autonomous mobile robot. In *Proceedings of 10th IEEE International Workshop on Robot and Human Interactive Communication*, pages 442–449, Bordeaux-Paris, France, September 2001.
- [46] R.O. Duba, P.E. Hart, and D.G. Stork. *Pattern Classification (Second Edition)*. A Wiley-Interscience Publication by John Wiley & Sons, INC., USA, 2001.
- [47] P.J. Escamilla-Ambrosio and N. Lieven. A multiple-sensor multiple-target tracking approach for the autotaxi system. In *Proceedings of 2004 IEEE Intelligent Vehicles Symposium*, pages 601–606, Parma, Italy, June 2004.
- [48] J.S. Evans and R.J. Evans. Image-enhanced multiple model tracking. *Automatica*, 35(11):1769–1786, November 1999.
- [49] T.J. Fan, G. Medioni, and G. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1140–1157, November 1989.
- [50] M.E. Farmer, R.-L. Hsu, and A.K. Jain. Interacting multiple model (imm) kalman filters for robust high speed human motion tracking. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 20–23, Quebec City, Canada, August 2002.
- [51] G. Farneback. Fast and accurate motion estimation using orientation tensors and parametric motion models. In *Proceedings of International Conference of Pattern Recognition*, volume I, pages 135–139, Barcelona, Spain, 2000.

- [52] G. Farneback. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume I, pages 171–177, Vancouver, Canada, July 2001.
- [53] J.A. Fayman, P. Pirjanian, H.I. Christensen, and E. Rivlin. Exploiting process integration and composition in the context of active vision. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 29:73–86, February 1999.
- [54] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, USA, 2002.
- [55] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the 13th International Conference on Uncertainty in Artificial Intelligence*, pages 175–181, San Francisco, CA, USA, August 1997.
- [56] A. Galata, N. Johnson, and D. Hogg. Learning variable-length markov models of behavior. *Computer Vision and Image Understanding*, 81(3):398–413, 2001.
- [57] P.J. Garcia-Padro, G.S. Sukhatme, and J.F. Montgomery. Toward vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous System*, 38(1):19–29, 2001.
- [58] W.F. Gardner and D.T. Lawton. Interactive model-based vehicle tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1115–1121, November 1996.
- [59] M. Gharavi-Alkhansari. A fast globally optimal algorithm for template matching using low-resolution pruning. *IEEE Transactions on Image Processing*, 10(4):526–533, April 2001.
- [60] H.H. Gonzalez-Banos, C.-Y. Lee, and J.-C. Latombe. Real-time combinatorial tracking of a target moving unpredictably among obstacles. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 2, pages 1683–1690, Washington, DC, USA, May 2002.

- [61] S.G. Goodridge, M.G. Kay, and R.C. Luo. Multilayered fuzzy behavior fusion for real-time reactive control of systems with multiple sensors. *IEEE Transactions on Industrial Electronics*, 43(3):387–394, June 2004.
- [62] B. Graf, M. Hans, and R.D. Schraft. Mobile robot assistants. *IEEE Robotics and Automation Magazine*, 2:67–77, June 2004.
- [63] H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic space-time video modeling via piecewise gmm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, March 2004.
- [64] M. Hagg and H.H. Nagel. Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision*, 35(3):295–319, 1999.
- [65] M.K. Hajjawi and A. Shirkhodaie. Cooperative visual team working and target tracking of mobile robots. In *Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory*, pages 376–380, Huntsville, AL, USA, March 2002.
- [66] D. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Boston, MA, USA, 1992.
- [67] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of IEEE*, 85:6–23, January 1997.
- [68] I. Haritaoglu, D. Harwood, and L.S. Davis. w^4 : Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:809–830, August 2000.
- [69] C.M. Higgins and V. Pant. A biomimetic vlsi sensor for visual tracking of small moving targets. *IEEE Transactions on Circuits and Systems I [see also IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications]*, 51:2384–2394, December 2004.
- [70] N. Hirai and H. Mizoguchi. Visual tracking of human back and shoulder for person following robot. In *Proceedings of the 2003 IEEE/ASME International Conference*

- on *Advanced Intelligent Mechatronics (AIM 2003)*, pages 527–532, Kobe, Japan, July 2003.
- [71] B.K.P. Horn. *Robot Vision*. MIT Press, USA, 1986.
- [72] J. Hoshino, H. Saito, and M. Yamamoto. A match moving technique for merging cg cloth and human video. *Journal of Visualization and Computer Animation*, 12(1):23–29, 2001.
- [73] W.M. Hu, T.N. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 34(3):334–352, August 2004.
- [74] C. Hue, J.-P. Le Cadre, and P. Prez. A particle filter to track multiple objects. In *Proceedings of IEEE Workshop on Multi-Object Tracking (WOMOT'01)*, pages 61–68, Vancouver, British Columbia, Canada, July 2001.
- [75] D.P. Huttenlocher, J.J. Noh, and W.J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Fourth International Conference on Computer Vision*, pages 93–101, Berlin, Germany, May 1993.
- [76] S. Huwer and H. Niemann. Adaptive change detection for real-time surveillance applications. In *Third IEEE International Workshop on Visual Surveillance*, pages 37–46, Dublin, Ireland, July 2000.
- [77] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of European Conference of Computer Vision*, pages 343–356, Cambridge, UK, April 1996.
- [78] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proceedings of European Conference of Computer Vision*, volume 1, pages 893–909, Cambridge, UK, April 1996.
- [79] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 28(1):5–28, 1998.
- [80] D.S. Jang and H.I. Choi. Active models for tracking moving objects. *Pattern Recognition*, 33(7):1135–1146, 2000.

-
- [81] G.J. Jang, S.H. Lee, and I.S. Kweon. Color landmark based self-localization for indoor mobile robots. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 1, pages 1037–1042, Washington, DC, USA, May 2002.
- [82] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *Proceedings of European Conference on Computer Vision*, volume 4, pages 343–357, Copenhagen, Denmark, May 2002.
- [83] B. Jensen, G. Froidevaux, X. Greppin, A. Lorotte, L. Mayor, M. Meisser, G. Ramel, and R. Siegwart. The interactive autonomous mobile system roblox. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 2, pages 1221–1227, Switzerland, October 2002.
- [84] M.H. Jeong, Y. Kuno, N. Shimada, and Y. Shirai. Two-hand gesture recognition using coupled switching linear model. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 529–532, Quebec City, Canada, August 2002.
- [85] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, October 2003.
- [86] Zhen Jia and A. Balasuriya. Motion based image segmentation with unsupervised bayesian learning. In *Proceedings of IEEE Workshop on Motion and Video Computing (WACV/MOTION’05)*, volume 2, pages 2–7, Colorado, USA, January 2005.
- [87] Zhen Jia and A. Balasuriya. Visual target tracking for autonomous vehicles with the optimal oosm solution. In *Proceedings of 2005 IEEE International Conference on Intelligent Transportation Systems*, pages 555–560, Vienna, Austria, September 2005.
- [88] Zhen Jia, A. Balasuriya, and S. Challa. Motion based 3d target tracking with interacting multiple linear dynamic models. In *Proceedings of the 15th British Machine Vision Conference*, United Kindom, September 2004.

- [89] Zhen Jia, A. Balasuriya, and S. Challa. Visual information and camera motion fusion for 3d target tracking. In *Proceedings of The Eighth International Conference on Control, Automation, Robotics and Vision*, pages 2296–2301, Kunming, China, December 2004.
- [90] Zhen Jia, A. Balasuriya, and S. Challa. Sensor fusion based 3d target visual tracking for autonomous vehicles with imm. In *Proceedings of 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 1841–1846, Barcelona, Spain, April 2005.
- [91] Zhen Jia, A. Balasuriya, and S. Challa. Target tracking with bayesian fusion based template matching. In *Proceedings of 2005 IEEE International Conference on Image Processing*, volume 2, pages 826–829, Genoa, Italy, September 2005.
- [92] N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 199–206, Kauai Marriott, Hawaii, USA, December 2001.
- [93] S. Ju, M.J. Black, and Y. Yaccob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 38–44, Los Alamitos, CA, USA, October 1996.
- [94] B.Y. Jung and G.S. Sukhatme. A region-based approach for cooperative multi-target tracking in a structured environment. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 3, pages 2764–2769, Switzerland, October 2002.
- [95] S.N. Kalitzin, J.J. Staal, B.M.terH. Romeny, and M.A. Viergever. Image segmentation and object recognition by bayesian grouping. In *Proceedings of 2000 IEEE International Conference on Image Processing*, volume 3, pages 580–583, Vancouver, BC, Canada, September 2000.
- [96] M. Kam, X.X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of IEEE*, 85(1):108–119, January 1997.

- [97] I.A. Karaulova, P.M. Hall, and A.D. Marshall. A hierarchical model of dynamics for tracking people with a single video camera. In *Proceedings of British Machine Vision Conference*, pages 252–262, Bristol, UK, September 2000.
- [98] H. Karlsson and J. Nygard. Robust and efficient tracking in image sequences using a kalman filter and an affine motion model. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 98–104, Switzerland, October 2002.
- [99] R.L. Kashyap and S. Sista. Bayesian framework for unsupervised classification with application to target tracking. In *Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1745–1748, Phoenix, AZ, USA, March 1999.
- [100] S. Katsura and K. Ohnishi. Human cooperative wheelchair for haptic interaction based on dual compliance control. *IEEE Transactions on Industrial Electronics*, 51(1):221–228, February 2004.
- [101] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 746–751, Kauai Marriott, Hawaii, USA, December 2001.
- [102] M. Kilger. A shadow handler in a video-based real-time traffic monitoring system. In *Proceedings of the IEEE Computer Society Workshop on Applications of Computer Vision*, pages 11–18, Palm Springs, CA, USA, November 1992.
- [103] K.R.S. Kodagoda. *Multi-Sensor Integration and Fusion in Mobile Robot Navigation*. PhD Degree Thesis, School of Electrical and Electronic Engineering, Nanyang Technological University, 2003.
- [104] M. Kohle, D. Merkl, and J. Kastner. Clinical gait analysis by neural networks: Issues and experiences. In *Proceedings of IEEE Symposium on Computer-Based Medical Systems*, pages 138–143, Maribor, Slovenia, June 1997.
- [105] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, New York:USA, 1988.

- [106] D. Koller, K. Daniilidis, and H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.
- [107] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Toward robust automatic traffic scene analysis in real-time. In *Proceedings of International Conference on Pattern Recognition*, pages 126–131, Jerusalem, Israel, October 1994.
- [108] H. Kollnig and H.H. Nagel. 3d pose estimation by directly matching polyhedral models to gray value gradients. *International Journal of Computer Vision*, 23(3):283–302, 1997.
- [109] A. Kosaka, M. Meng, and A.C. Kak. Vision-guided mobile robot navigation using retroactive updating of position uncertainty. In *Proceedings of 1993 IEEE International Conference on Robotics and Automation*, volume 2, pages 1–7, Atlanta, Georgia, USA, May 1993.
- [110] Y. Kuno, T. Watanabe, Y. Shimosakoda, and S. Nakagawa. Automated detection of human for visual surveillance system. In *Proceedings of International Conference on Pattern Recognition*, volume 3, pages 865–869, Vienna, Austria, August 1996.
- [111] H.K. Kwon, Y.R. Yoon, J.B. Park, and A.C. Kak. Person tracking with a mobile robot using two uncalibrated independently moving cameras. In *Proceedings of 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.
- [112] F. Lamiraux, D. Bonnafeux, and O. Lefebvre. Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 20(6):967–977, December 2004.
- [113] C.-Y. Lee, H. Gonzalez-Banos, and J.-C. Latombe. Real-time tracking of an unpredictable target amidst unknown obstacles. In *Proceedings of Seventh International Conference on Control, Automation, Robotics And Vision (ICARCV'02)*, volume 4, pages 596–601, Singapore, December 2002.

-
- [114] J.-H. Lee, K. Morioka, and H. Hashimoto. Physical distance based human robot interaction in intelligent environment. In *Proceedings of IEEE 2002 28th Annual Conference of the IECON [Industrial Electronics Society]*, volume 4, pages 3068–3073, Sevilla, Spain, November 2002.
 - [115] I. Leichter, M. Lindenbaum, and E. Rivlin. A probabilistic framework for combining tracking algorithms. In *Proceedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 445–451, Washington, DC, USA, July 2004.
 - [116] K. Li, X.D. Wu, D.Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images—a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:119–134, January 2005.
 - [117] M.K. Li, I.K. Sethi, D.G. Li, and N. Dimitrova. Region growing using online learning. In *Proceedings of the International Conference on Imaging Science, Systems and Technology, CISST '03*, pages 73–76, Las Vegas, Nevada, USA, June 2003.
 - [118] S. Li and S. Tsuji. Finding landmarks autonomously along a route. In *Proceedings of the 11 IAPR International Conference on Computer Vision and Applications*, pages 316–319, Hague, Netherlands, September 1992.
 - [119] T.-H.S. Li, S.-J. Chang, and W. Tong. Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. *IEEE Transactions on Fuzzy Systems*, 12(4):491–501, August 2004.
 - [120] J.W. Lim, D. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, L. Saul, Y. Weiss, and L. Bottou (eds), 2005.
 - [121] R.-S. Lin, D. Ross, J.W. Lim, and M.-H. Yang. Adaptive discriminative generative model and its applications. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, L. Saul, Y. Weiss, and L. Bottou (eds), 2005.
 - [122] A.J. Lipton. Local application of optical flow to analyze rigid versus nonrigid motion. In *Proceedings of International Conference on Computer Vision Workshop on Frame-Rate Vision*, Corfu, Greece, 1999.

- [123] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. In *Proceedings of IEEE International Workshop on Applications of Computer Vision*, pages 8–14, Princeton, New Jersey, USA, October 1998.
- [124] F. Liu, X.Y. Lin, S.Z. Li, and Y. Shi. Multi-modal face tracking using bayesian network. In *Proceedings of IEEE International Workshop on Analysis and Modelling of Faces and Gestures*, pages 135–142, Nice, France, October 2003.
- [125] H.Y. Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation using adaptive structure tensors and a parametric model. *IEEE Transactions on Image Processing*, 12(10):1170–1180, October 2003.
- [126] J.G. Lou, H. Yang, W.M. Hu, and T.N. Tan. Visual vehicle tracking using an improved ekf. In *Proceedings of Asian Conference on Computer Vision*, pages 296–301, Melbourne, Australia, January 2002.
- [127] D.G. Lowe. Fitting parameterized 3-d models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.
- [128] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo. In *Proceedings of DARPA Image Understanding Workshop*, pages 121–130, April 1981.
- [129] R.C. Luo, C.C. Yih, and K.L. Su. Multisensor fusion and integration: Approaches, applications, and future research directions. *IEEE Sensors Journal*, 2(2):107–119, April 2002.
- [130] J. Malik and S. Russell. Traffic surveillance and detection technology development: New traffic sensor technology. *University of California, Berkeley, California PATH Research Final Report, UCB-ITS-PRR-97-6*, 1997.
- [131] A. Marin-Hernandez, V. Ayala-Ramireze, and M. Devy. Collaborative tracking for robotic tasks. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 272–277, Las Vegas, Nevada, USA, October 2003.

- [132] C. Marques and P. Lima. Avoiding obstacles-multisensor navigation for nonholonomic robots in cluttered environments. *IEEE Robotics and Automation Magazine*, 11:70–82, September 2004.
- [133] I. Matthews, T. Ishikawaa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810–815, June 2004.
- [134] L. Mattos and E. Grant. Passive sonar applications: target tracking and navigation of an autonomous robot. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 5, pages 4265–4270, New Orleans, LA, USA, May 2004.
- [135] M. Jr. Mazo, A. Speranzon, K.H. Johansson, and X.M. Hu. Multi-robot tracking of a moving object using directional sensors. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, volume 2, pages 1103–1108, New Orleans, LA, USA, May 2004.
- [136] S.J. Mckenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.
- [137] D. Meyer, J. Denzler, and H. Niemann. Model based extraction of articulated objects in image sequences for gait analysis. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 78–81, Washington, DC, USA, October 1997.
- [138] D. Meyer, J. Psl, and H. Niemann. Gait classification with hmm’s for trajectories of body parts extracted by mixture densities. In *Proceedings of the British Machine vision Conference*, pages 459–468, Southampton, UK, September 1998.
- [139] R. Miller, B. Mettler, and O. Amidi. Carnegie mellon university’s 1997 international aerial robotics competition entry. In *Proceedings of International Aerial robotics Competition*, 1997.
- [140] T. Mitchell. *Machine Learning*. McGraw Hill, USA, 1997.

-
- [141] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.
- [142] R. Mori and F. Miyazaki. Gag (gaining angel of gaze) strategy for ball tracking and catching task. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 1, pages 281–286, Switzerland, October 2002.
- [143] S. Mori, H. Nishida, and H. Yamada. *Optical Character Recognition*. John Wiley & Sons, USA, 1999.
- [144] K. Morioka, J.H. Lee, and H. Hashimoto. Physical agent for human following in intelligent sensor network. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 2, pages 1234–1239, Switzerland, October 2002.
- [145] K. Morioka, J.H. Lee, and H. Hashimoto. Human-following mobile robot in a distributed intelligent sensor network. *IEEE Transactions on Industrial Electronics*, 51(1):229–237, February 2004.
- [146] K. Muhlmann, D. Maier, R. Hesser, and R. Manner. Calculating dense disparity maps from color stereo images, an efficient implementation. In *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 30–36, Kauai, Hawaii, USA, December 2001.
- [147] R. Murrieta, A. Sarmiento, and S. Hutchinson. On the existence of a strategy to maintain a moving target within the sensing range of an observer reacting with delay. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1184–1191, Las Vegas, Nevada, USA, October 2003.
- [148] R. Murrieta-Cid, M. Briot, and N. Vandapel. Landmark identification and tracking in natural environment. In *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 179–184, Victoria, Canada, October 1998.

- [149] S. Nadimi and B. Bhanu. Multistrategy fusion using mixture model for moving object detection. In *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 317–322, Baden-Baden, German, August 2001.
- [150] W.K. Ng, G.S.B. Leng, and Y.L. Low. Coordinated movement of multiple robots for searching a cluttered environment. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 400–405, Sendai, Japan, October 2004.
- [151] H.T. Nguyen and A.W.M. Smeulders. Fast occluded object tracking by a robust appearance filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1099–1104, August 2004.
- [152] H.Z. Ning, L. Wang, W.M. Hu, and T.N. Tan. Articulated model based people tracking using motion models. In *Proceedings of the IEEE International Conference on Multi-Model Interfaces*, pages 383–388, Pittsburgh, USA, October 2002.
- [153] K. Nishiwaki, S. Kagami, J.J. Kuffner, M. Inaba, and H. Inoue. Online humanoid walking control system and a moving goal tracking experiment. In *Proceedings Of the 2003 IEEE International Conference on Robotics and Automation*, pages 911–916, Taipei, Taiwan, September 2003.
- [154] S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 469–474, Seattle, WA, USA, June 1994.
- [155] T. Ogata, S. Matsuda, J.K. Tan, and S. Ishikawa. Real-time human motion recognition by aerial robot. In *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 6, pages 5290–5295, Hague, Netherlands, October 2004.
- [156] P. Ogren and N.E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics*, 21(2):188–195, April 2005.
- [157] R. Okada, Y. Shirai, and J. Miura. Object tracking based on optical flow and depth. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and*

- Integration for Intelligent Systems*, pages 565–571, Washington D.C., USA, December 1996.
- [158] R. Okada, Y. Shirai, and J. Miura. Tracking a person with 3-d motion by integrating optical flow and depth. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 336–341, Grenoble, France, March 2000.
- [159] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, March 2002.
- [160] C.A. Pau, A. Barber, K. Kennedy, E. Olejniczak, J. Hackney, Q. Sun, and E. Mireles. Traffic sensor using a color vision method. In *Proceedings of SPIE-Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, volume 2902, pages 156–165, Boston, Massachusetts, USA.
- [161] V. Pavlovic, J.M. Rehg, T.J. Cham, and K.P. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamical models. In *Proceedings of International Conference on Computer Vision*, pages 94–101, Kerkyra, Corfu, Greece, September 1999.
- [162] A.E.C. Pece and A.D. Worrall. Tracking without feature detection. In *Proceedings of IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 29–37, Grenoble, France, March 2000.
- [163] H. Permuter, J. Francos, and I.H. Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *Proceedings of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing(ICASSP '03)*, volume 3, pages 569–572, Hong Kong, China, April 2003.
- [164] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):564–569, June 1999.

- [165] R. Plankers and P. Fua. Articulated soft objects for video-based body modelling. In *Proceedings of IEEE Computer Society International Conference on Computer Vision*, volume 1, pages 394–401, Vancouver, BC, Canada, July 2001.
- [166] R. Polana and R. Nelson. Low level recognition of human motion (or how to get your man without finding his body parts). In *Proceedings of IEEE Computer Society International Workshop on Non-Rigid and Articulated Objects*, pages 77–82, Austin, TX, USA, November 1994.
- [167] Z.H. Qu, J. Wang, and C.E. Plaisted. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, 20(6):978–993, December 2004.
- [168] Y. Raja, S.J. McKenna, and S.D. Gong. Segmentation and tracking using colour mixture models. In *Proceedings of Asian Conference on Computer Vision*, pages 607–614, Hong Kong, January 1998.
- [169] Y. Raja, S.J. McKenna, and S.G. Gong. Colour model selection and adaptation in dynamic scenes. In *Proceedings of European Conference of Computer Vision (ECCV)*, pages 460–474, Freiburg, Germany, June.
- [170] S. Ray and R.H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, pages 137–143, Calcutta, India, December 1999.
- [171] R.E. Woods R.C. Gonzalez. *Digital Image Processing (Second Edition)*. Prentice Hall, USA, 2001.
- [172] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman filtering. In *Proceedings of the International Conference on Recent Advances in Mechatronics*, pages 193–199, Istanbul, Turkey, August 1995.
- [173] M. Ridley, E. Nettleton, S. Sukkarieh, and H. Durrant-Whyte. Tracking in decentralised air-ground sensing networks. In *Proceedings of the Fifth International*

- Conference on Information Fusion*, volume 1, pages 616–623, Annapolis, USA, July 2002.
- [174] P. Rives and J.-J. Borrelly. Underwater pipe inspection task using visual servoing techniques. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1997. IROS '97.*, volume 1, pages 63–68, Genoble, France, September 1997.
- [175] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, January 1994.
- [176] D. Ross, J.W. Lim, and M.-H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proceedings of the Eighth European Conference on Computer Vision (ECCV 2004)*, volume 2, pages 470–482, Prague, Czech Republic, May 2004.
- [177] S. Saripalli, J.F. Montgomery, and G.S. Sukhatme. Vision based autonomous landing of an unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2799–2804, Washington, D.C., US, May 2002.
- [178] S. Saripalli, J.F. Montgomery, and G.S. Sukhatme. Visually guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380, June 2003.
- [179] S. Saripalli and G.S. Sukhatme. Landing on a moving target using an autonomous helicopter. In *Proceedings of the International Conference on Field and Service Robotics*, Mt Fuji, Japan, July 2003.
- [180] B. Schiele. Model-free tracking of cars and people based on color regions. In *Proceedings of IEEE Computer Society International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 61–71, Grenoble, France, 2000.
- [181] G.R. Scholz and C.D. Rahn. Profile sensing with an actuated whisker. *IEEE Transactions on Robotics and Automation*, 20(1):124–127, February 2004.

- [182] O. Shakernia, Y. Ma, T.J. Koo, and S.S. Sastry. Landing an unmanned air vehicle: vision based motion estimation and non-linear control. *Asian Journal of Control*, 1:128–145, September 1999.
- [183] O. Shakernia, R. Vidal, C.S. Sharp, Y. Ma, and S.S. Sastry. Multiple-view motion estimation and control for landing an unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2793–2798, Washington, DC, USA, May 2002.
- [184] T. Shamir. How should an autonomous vehicle overtake a slower moving vehicle: Design and analysis of an optimal trajectory. *IEEE Transactions on Automatic Control*, 49(4):607–610, April 2004.
- [185] C.S. Sharp, O. Shakernia, and S.S. Sastry. A vision system for landing an unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1720–1727, Seoul, Korea, 2001.
- [186] H.Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *Proceedings of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition*, pages 427–433, Santa Barbara, CA, USA, June 1998.
- [187] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proceedings of The European Conference on Computer Vision*, volume 2, pages 702–718, Dublin, Ireland, June 2000.
- [188] R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proceedings of 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1060–1065, Victoria, Canada, October 1998.
- [189] B. Sinopoli, M. Micheli, G. Donato, and T.J. Koo. Vision based navigation for an unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1757–1764, Seoul, Korea, 2001.
- [190] S. Sista and R.L. Kashyap. Bayesian estimation for multiscale image segmentation. In *Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3493–3496, Phoenix, AZ, USA, March 1999.

-
- [191] K. Skrzypczyk. Game theory based target following by a team of robots. In *Proceedings of the Fourth International Workshop on Robot Motion and Control*, pages 91–96, Puzszykowo, Poland, June 2004.
- [192] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 447–454, Kauai, HI, USA, 2001.
- [193] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):479–494, April 2004.
- [194] C. Stauffer. Automatic hierarchical classification using time-base co-occurrences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2333–2339, Ft. Collins, CO, USA, June 1999.
- [195] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2246–2252, Ft. Collins, CO, USA, June 1999.
- [196] S. Stolle and R. Rysdyk. Flight path following guidance for unmanned air vehicles with pan-tilt camera for target observation. In *Proceedings of The 22nd Digital Avionics Systems Conference*, volume 2, pages 8.B.3–81–12, Indianapolis, Indiana, USA, October 2003.
- [197] J.A. Stover, D.L. Hall, and R.E. Gibson. A fuzzy-logic architecture for autonomous multisensor data fusion. *IEEE Transaction on Industry Electronics*, 43:403–410, June 1996.
- [198] E. Stringa. Morphological change detection algorithms for surveillance applications. In *Proceedings of British Machine Vision Conference*, pages 402–411, Bristol, UK, 2000.

-
- [199] H.Z. Sun, T. Feng, and T.N. Tan. Robust extraction of moving objects from image sequences. In *Proceedings of Asian Conference on Computer Vision*, pages 961–964, Taiwan, January 2000.
- [200] P. Sykacek, S.J. Roberts, and M. Stokes. Adaptive bci based on variational bayesian kalman filtering: An empirical evaluation. *IEEE Transactions on Biomedical Engineering*, 51(5):719–729, May 2004.
- [201] T.N. Tan and K.D. Baker. Efficient image gradient based vehicle localization. *IEEE Transaction on Image Processing*, 9:1343–1356, August 2000.
- [202] T.N. Tan, G.D. Sullivan, and K.D. Baker. Fast vehicle localization and recognition without line extraction. In *Proceedings of British Machine Vision Conference*, pages 95–104, Birmingham, UK, September 1994.
- [203] T.N. Tan, G.D. Sullivan, and K.D. Baker. Pose determination and recognition of vehicles in traffic scenes. In *Proceedings of European Conference on Computer Vision-Lecture Notes in Computer Science*, volume 1, pages 501–506, Stockholm, Sweden, May 1994. J.O. Eklundh, Editions.
- [204] T.N. Tan, G.D. Sullivan, and K.D. Baker. Model-based localization and recognition of road vehicles. *International Journal of Computer Vision*, 27(1):22–25, 1998.
- [205] H. Tao, H.S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 134–141, Hilton Head Island, South Carolina, USA, June.
- [206] S.C.A. Thomopoulos. Sensor integration and data fusion. *Journal of Robotic System*, 7(3):337–372, 1990.
- [207] T.Y. Tian, C. Tomasi, and D.J. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 315–320, San Francisco, CA, USA, June 1996.

- [208] M. Tomono. Path planning with target finding by single camera under map uncertainty. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 465–470, Kobe, Japan, June 2003.
- [209] E. Valveny and E. Marti. Hand-drawn symbol recognition in graphic documents using deformable template matching and a bayesian framework. In *Proceedings of 15th International Conference on Pattern Recognition*, volume 2, pages 239–242, Barcelona, Spain, September 2000.
- [210] J. Vass, K. Palaniappan, and X.H. Zhuang. Automatic spatio-temporal video sequence segmentation. In *Proceedings of 1998 International Conference on Image Processing*, volume 1, pages 958–962, Chicago, Illinois, USA, October 1998.
- [211] J. Vermaak, P. Perez, M. Gangnet, and A. Blake. Towards improved observation models for visual tracking, selective adaptation. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 645–660, Copenhagen, Denmark, May 2002.
- [212] S. Wachter and H.H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, June 1999.
- [213] C.-C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, volume 3, pages 2918–2924, Washington, DC, USA, May 2002.
- [214] H. Wang, T. Kirubarajan, Y. Li, and Y. Bar-shalom. Precision large scale air traffic surveillance using imm/assignment estimators. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):255–266, January 1999.
- [215] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.
- [216] Y. Wang, T.L. Tan, and K.-F. Loe. Video segmentation based on graphical models. In *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 335–342, Madison, Wisconsin, USA, June 2003.

-
- [217] R. Wei, R. Mahony, and D. Austin. A bearing-only control law for stable docking of unicycles. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 3793–3798, Las Vegas, USA, October 2003.
 - [218] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.
 - [219] Y. Wu and T.S. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *International Journal of Computer Vision*, 58(1):55–71, June 2004.
 - [220] S. Yamamoto, Y. Mae, Y. Shirai, and J. Miura. Realtime multiple object tracking based on optical flows. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 3, pages 2328–2333, Nagoya, Aichi, Japan, May 1995.
 - [221] T. Yamane, Y. Shirai, and J. Miura. Person tracking by integrating optical flow and uniform brightness regions. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3267–3272, Leuven, Belgium, May 1998.
 - [222] F. Yang. *Vision-Based Target Tracking and Navigation System for Autonomous Underwater Vehicles*. Master Degree Thesis, School of Electrical and Electronic Engineering, Nanyang Technological University, 2001.
 - [223] F. Yang and A. Balasuriya. Optical flow based speed estimation in auv target tracking. In *Proceedings of MTS/IEEE Conference and Exhibition on OCEANS*, volume 4, pages 2377–2382, Hawaii, USA, November.
 - [224] F.G. Yang and T.Z. Jiang. Pixon-based image segmentation with markov random fields. *IEEE Transactions on Image Processing*, 12(12):1552–1559, December 2003.
 - [225] H. Yang, J.G. Lou, H.Z. Sun, W.H. Hu, and T.N. Tan. Efficient and robust vehicle localization. In *Proceedings of IEEE International Conference on Image Processing*, volume 2, pages 355–358, Thessaloniki, Greece, October 2001.
-

-
- [226] M.H. Yang. *Hand gesture recognition and face detection in image*. Ph.D. thesis, University of Illinois at Urbana-Champaign, 2002.
- [227] S.X. Yang and C. Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 34(1):718–724, February 2004.
- [228] J.Z. Yu, S. Wang, and M. Tan. A parallel algorithm for visual tracking of multiple free-swimming robot fishes based on color information. In *Proceedings of 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, volume 1, pages 359–364, Changsha, China, October 2003.
- [229] Q. Yu, M. Yang, and H. Wang. Real-time vision based ego-motion estimation for outdoor mobile robot. In *Proceedings of the 5th World Congress on Intelligent Control and Automation*, volume 6, pages 4720–4724, Hangzhou, China, June 2004.
- [230] H.B. Zhang, K. Yuan, and J.-D. Liu. A fast and robust vision system for autonomous mobile robots. In *Proceedings of 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, volume 1, pages 60–65, Changsha, China, October 2003.
- [231] Y.N. Zhang and J. Wang. Obstacle avoidance for kinematically redundant manipulators using a dual neural network. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 34(1):752–759, February 2004.
- [232] Z.Y. Zhang. Modelling geometric structure and illumination variation of a scene from real images. In *Proceedings of IEEE Conference on Computer Vision*, pages 1041–1046, Bombay, India, January 1998.
- [233] T. Zhao, T.S. Wang, and H.Y. Shum. Learning a highly structured motion model for 3d human tracking. In *Proceedings of Asian Conference on Computer Vision*, pages 144–149, Melbourne, Australia, January 2002.
- [234] Y. Zhong, A.K. Jain, and M.-P. Dubuisson-Jolly. Object tracking using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):544–549, May 2000.

-
- [235] S.H.K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13(11):1491–1506, November 2004.