Hybrid sensor fusion for unmanned ground vehicle

Guan, Mingyang

2020

Guan, M. (2020). Hybrid sensor fusion for unmanned ground vehicle. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/144485>

<https://doi.org/10.32657/10356/144485>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 19:34:47 SGT

HYBRID SENSOR FUSION FOR UNMANNED GROUND VEHICLE

MINGYANG GUAN

School of Electrical & Electronic Engineering

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

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

25-03-2020

.....

Date

Guan Mingyang

.....

Mingyang Guan

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

25-03-2020

.....
Date

.....
Changyun Wen

Authorship Attribution Statement

This thesis contains material from 2 papers published in the following peer-reviewed journal and conference, and 3 papers under reviewing where I was the first or joint first authors.

Chapter 3 is published as Guan, M., Wen, C., Shan, M., Ng, C. L., and Zou, Y. Real-time event-triggered object tracking in the presence of model drift and occlusion. *IEEE Transactions on Industrial Electronics*, 66(3), 2054-2065 (2018). DOI: 10.1109/TIE.2018.2835390.

The contributions of the co-authors are as follows:

- Prof. Wen provided the initial idea.
- I designed the event-triggered tracking framework and the algorithm of online object relocation.
- I co-designed the study with Prof Wen and performed all the experimental work at the ST-NTU Corplab. I also analyzed the data.
- Mr. Ng and Ms. Zhou assisted to collect the experimental results.
- I prepared the manuscript drafts. The manuscript was revised together with Prof. Wen and Dr. Shan.

Chapter 4 is accepted as Guan, M., and Wen, C. Adaptive Multi-feature Reliability Re-determination Correlation Filter for Visual Tracking. *IEEE Transactions on Multimedia*.

The contributions of the co-authors are as follows:

- Prof. Wen provided the initial idea.
- I co-designed the study with Prof Wen and performed all the experimental work at the ST-NTU Corplab.
- I proposed the tracking framework and two solutions on finding the reliability score of each feature.
- I prepared the manuscript drafts which were revised by Prof Wen.

Chapter 5 is published as Song, Y^{*}, Guan, M^{*}, Tay, W.P., Law, C.L. and Wen, C., UWB/LiDAR Fusion For Cooperative Range-Only SLAM. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6568-6574, 2019 May. DOI: 10.1109/ICRA.2019.8794222. **The authors with ^{*} are the jointly first author of this publication.**

The contributions of the co-authors are as follows:

- Prof. Wen suggested the idea on the fusion of UWB/LiDAR.
- I wrote the drafts related to LiDAR SLAM, and Dr. Song prepared the drafts related to UWB localization.
- I co-designed the fusion framework with Dr. Song.
- I designed and implemented all the experiments for the proposed method. The manuscript was revised together with Prof. Wen and Dr. Song.
- Dr. Song implemented the experiments related to the UWB localization.
- Prof. Tay and Prof. Law provided the advices on UWB sensors.

Chapter 6 is under reviewing as Guan, M., Wen, C., Song, Y., and Tay, W.P., Autonomous Exploration Using UWB and LiDAR. *IEEE Transactions on Industrial Electronics*.

The contributions of the co-authors are as follows:

- Prof. Wen suggested the idea of fusing UWB/LiDAR for autonomous exploration.
- I proposed a particle filter based step-by-step optimization framework to refine the state of robot and UWB beacons.
- I prepared the manuscript drafts. The manuscript was revised together with Prof. Wen and Dr. Song.
- Prof. Tay provided the advice on UWB sensors.
- I co-designed the study with Prof Wen and performed all the experimental work at the ST-NTU Corplab.

Chapter 7 is under reviewing as Guan, M., Wen, C., and Song, Y. Autonomous Exploration via Region-aware Least-explored Guided Rapidly-exploring Random Trees. *Journal of Field Robotics*.

The contributions of the co-authors are as follows:

- Prof. Wen suggested the idea of fusion UWB/LiDAR for autonomous exploration.
- I proposed a dual-UWB robot system and a least-explored guided RRTs for autonomous exploration.
- I prepared the manuscript drafts. The manuscript was revised together with Prof. Wen and Dr. Song.
- I co-designed the study with Prof Wen and performed all the experimental work at the ST-NTU Corplab.

25-03-2020

.....

Date

Guan Mingyang

.....

Mingyang Guan

Acknowledgements

First of all, I wish to express my greatest gratitude and deepest appreciation to my advisor, Prof. Changyun Wen, for his continuous support, professional guidance and sincere encouragement throughout my PhD study. Prof. Wen's serious scientific attitude, rigorous scholarship and optimistic outlook on life, would always inspire me to work harder and live happier in the future. This thesis would not be possible without his brilliant ideas and extraordinary drive for research.

Secondly, I would like to express my special thanks to Dr. Mao Shan, Dr. Zhe Wei, Dr. Zhengguo Li and Dr. Yang Song for their instructions, encouragement and assistance in my Ph.D research. When I began my Ph.D study, Mao helped me to quickly get familiar with the important technologies involving robotics. He always discussed with me patiently to help me find solutions when I encountered problems. After Mao left NTU, Zhe has helped me to conquer some hard issues of the "Smart Wheelchair" project and analysis the experimental results. Later, Zhengguo has guided me a lot on problem formulating and solving, he also shared me some valuable knowledges and research directions. During my last years of Ph.D study, Yang has helped me in both theoretical and experimental studies. We have had pleasant and encouraging discussions about both the project and my Ph.D study. Overall, their kind supports helped me overcome lots of difficulties during my PhD study.

Thirdly, I want to thank my colleagues and friends, Dr. Xiucai Huang, Dr. Renjie He, Dr. Fanghong Guo, Dr. Fei Kou, Ms. Ying Zou, Dr. Lantao Xing, Mr. Ruibin Jing, Dr. Jie Ding, Dr. Jingjing Huang and Dr. Hui Gao who are in Prof. Wen's group, Dr. Yuanzhe Wang, Mr. Chongxiao Wang, Mr. Yijie Zeng, Mr. Mok Bo Chuan, Dr. Yunyun Huang, Mr. Yan Xu, Mr. Kok Ming Lee, Mr. Paul Tan, Mr. Pek Kian Huat Alex and Mr. Song Guang Ho who are in ST Engineering-NTU Corporate Laboratory, for the experience of studying and working with them, and also for their generous help in countless experiments.

Last but not least, I would like to express my deepest thanks to my parents, my wife as well as other family members, for their endless love and unswerving support.

Abstract

The unmanned ground vehicles (UGVs) have been applied to execute many important tasks in the real world scenarios such as surveillance, exploring the hazard environment and autonomous transportation. The UGV is a complex system as it is integrated by several challenge technologies, such as simultaneously localization and mapping (SLAM), collision-free navigation, and robotic perception. Generally, the navigation and control of UGVs in the Global Positioning System (GPS) denied environment (i.e., indoor scenario) are critically dependent on the SLAM system which provides localization service for UGVs, while the robotic perception endows UGVs the ability of understanding their surrounding environments, such as continuously tracking the moving obstacles and then filtering them out in the localization process. In this thesis, we concentrate on the two topics involving autonomously robotic systems, say SLAM and visual object tracking.

The first part of this thesis focuses on visual object tracking, which is to generally estimate the motion state of the given target based on its appearance information. Though many promising tracking models have been proposed in the recent decade, some challenges are still waiting to be addressed, such as computational efficiency, tracking model drift due to illumination variation, motion blur, occlusion and deformation. Therefore, we address these issues by proposing two trackers: 1) Event-triggered tracking (ETT) framework which attempts to enable the tracking task to be carried out by an efficient short-term tracker (i.e., correlation filter based tracker) in most of the time while triggers to restore the short-term tracker once it fails to track the target, thus a balance between tracking accuracy and efficiency is achieved; 2) reliability re-determinative correlation filter (RRCF) which aims to take advantages from multiple feature representations to robustify the tracking model. Meanwhile, we propose two different weight solvers to adaptively adjust the importance of each feature. Extensive experiments have been designed on several large datasets to validate that: 1) the proposed tracking framework is superior to enhance the robustness of tracking model, 2) the proposed two weight solvers can

effectively find the optimal weight for each feature. As expected, the proposed two trackers indeed improve the accuracy and robustness compared to the state-of-the-art trackers. Especially on VOT2016, the proposed RRCF achieves an outstanding EAO scores of **0.453**, which outperforms the recent top trackers by a large margin.

The second part of this thesis considers the issue of SLAM. Typically, the SLAM system relies on the information collected from the sensors, like LiDAR, camera and IMU, which either suffers from accumulated localization error due to the lack of global reference, or requires more time to detect the loop closure yet reduces the efficiency. To handle the issue of error accumulation, we propose to integrate several low-cost radio frequency technology based sensors (i.e., ultra-wideband (UWB)) and LiDAR/Camera to construct a fusion SLAM for the GPS-denied environment. We propose to fuse the peer-to-peer ranges measured among UWB nodes and laser scanning information, i.e. range measured between robot and nearby objects/obstacles, for simultaneous localization of the robot, all UWB beacons and LiDAR mapping. The fusion is inspired by two facts: 1) LiDAR may improve UWB-only localization accuracy as it gives a more precise and comprehensive picture of the surrounding environment; 2) on the other hand, UWB ranging measurements may remove the error accumulated in the LiDAR-based SLAM algorithm. More importantly, two different fusion schemes, named one-step optimization¹ and step-by-step optimization^{2,3}, are proposed in this thesis to tightly fuse UWB ranges with LiDAR scanning. The experiments demonstrate that UWB/LiDAR fusion enables drift-free SLAM in real-time based on ranging measurements only.

Furthermore, since the established UWB/LiDAR fusion SLAM system not only provide drift-free localization service for UGVs, but also sketch an abstract map (i.e., to-be-explored region) about the environment, a fully autonomous exploration system^{2,3} is built upon a UWB/LiDAR fusion SLAM. A where-to-explore scheme is proposed to guide the robot to the less-explored areas, which is implemented together with a collision-free navigation system and global path planning module. With such modules, the robot is endowed with ability of autonomously exploring an environment and build the detailed map for it. In the navigation process, we use UWB beacons, whose locations are estimated on the fly, to sketch the region where the robot is going to explore. In the process of mapping, UWB sensors

¹video in workshop: <https://youtu.be/yZIK37ykTGI>

²video in workshop: https://youtu.be/degpuH_h2AM

³video in garden: <https://youtu.be/FQQBuIuid2s>

equipped on the robot provide real-time location estimates which help remove the accumulated errors in the LiDAR-only SLAM. Experiments are conducted in two different environments, a cluttered workshop and a spacious garden, to verify the effectiveness of our proposed strategy. The experimental tests involving UWB/LiDAR fusion SLAM and autonomous exploration are filmed ^{2,3}.

Contents

Acknowledgements	xi
Abstract	xiii
List of Figures	xxi
List of Tables	xxv
Symbols and Acronyms	xxvii
1 Introduction	1
1.1 Motivation and Objectives	3
1.1.1 Visual Object Tracking	3
1.1.1.1 Robust Visual Object Tracking via Event-Triggered Tracking Failure Restoration	5
1.1.1.2 Robust Visual Object Tracking via Multi-feature Reliability Re-determination	7
1.1.2 Sensor Fusion for SLAM	9
1.1.3 UWB/LiDAR Fusion for Autonomous Exploration	14
1.2 Major Contributions	18
1.3 Organization of the Thesis	21
2 Literature Review	23
2.1 Visual Object Tracking	23
2.1.1 Tracking-by-detection	23
2.1.2 Correlation Filter Based Tracking	24
2.1.3 Deep Learning Based Tracking	26
2.1.4 Trackers with Model Drift Alleviation	27
2.1.5 Ensemble Tracking	28
2.2 Simultaneous Localization and Mapping (SLAM)	30
2.2.0.1 Wireless Sensor based SLAM	30
2.2.0.2 LiDAR based SLAM	31
2.2.0.3 Vision based SLAM	32
2.2.0.4 Sensor fusion for SLAM	33

2.3	Robotic Exploration System	34
2.3.1	Single-Sensor based Autonomous Exploration	35
2.3.2	Autonomous Exploration with Background Information	36
3	Event-Triggered Object Tracking for Long-term Visual Tracking	39
3.1	Introduction	39
3.1.1	Technical Comparisons Among Similar Trackers	41
3.2	Existing Techniques Used in Our Proposed Tracker	42
3.2.1	Techniques in Correlation Filter-based Tracking	42
3.2.2	Techniques in Discriminative Online-SVM Classifier	44
3.3	The Proposed Techniques	46
3.3.1	Occlusion and Model Drift Detection	46
3.3.1.1	Spatial Loss Evaluation	47
3.3.1.2	Temporal Loss Evaluation	48
3.3.2	Event-triggered Decision Model	49
3.3.2.1	Correlation Tracking Model Updating	51
3.3.2.2	Heuristic Object Re-detection	52
3.3.2.3	Detector Model Updating	54
3.3.2.4	Re-sampling for Detector Model	56
	Normal Tracking	56
3.4	Experiments	57
3.4.1	Implementation Details	57
3.4.2	Evaluation on OTB-50 [1] and OTB-100 [2] Dataset	59
3.4.2.1	Components Analysis	60
3.4.2.2	Quantitative Evaluation	61
3.4.2.3	Comparisons on different attributes on OTB-100	62
3.4.2.4	Comparisons on tracking speed	62
3.4.2.5	Evaluation on VOT16 [3] Dataset	63
3.4.3	Discussion on Failure Cases	63
3.5	Conclusions	65
4	Adaptive Multi-feature Reliability Re-determination Correlation Filter for Visual Tracking	67
4.1	Introduction	67
4.2	Reliability Re-determination Correlation Filter	68
4.2.1	Estimation of the CFs	69
4.2.2	Estimation of the Reliability $w_{t,k}$	71
4.2.2.1	Estimating $w_{t,k}$ through Numerical Optimization	71
4.2.2.2	Estimating $w_{t,k}$ through Model Evaluation	74
4.3	Experimental Tests and Results	78
4.3.1	Implementation Details	78
4.3.2	Tracking Framework Study	78
4.3.2.1	Tracking with different features	78
4.3.2.2	Evaluation on the effects of $\mathbf{w}_t$	79

4.3.2.3	Parameter selection	80
4.3.3	Comparison with state-of-the-art trackers	81
4.3.3.1	Evaluation on OTB-50 and OTB-100	81
4.3.3.2	Evaluation on TempleColor	84
4.3.3.3	Evaluation on LaSOT	85
4.3.3.4	Evaluation on VOT2016	86
4.3.3.5	Evaluation on VOT2018	87
4.3.4	Comparison between the proposed two solutions	88
4.3.5	Comparisons on different attributes on OTB-100	88
4.3.6	Qualitative evaluation	89
4.3.7	Analysis on the tracking speed	89
4.3.8	Discussion on pros and cons	90
4.4	Conclusion	91
5	UWB/LiDAR Fusion SLAM via One-step Optimization	93
5.1	Introduction	93
5.2	Problem Formulation	94
5.3	UWB-Only SLAM	95
5.3.1	The dynamical and observational models	95
5.3.2	EKF update	96
5.3.3	Elimination of location ambiguities	97
5.3.4	Detection/removal of ranging outliers	98
5.4	Map Update	98
5.5	Experimental Results	102
5.5.1	Hardware	102
5.5.2	SLAM in a workshop of size 12×19 m ²	103
5.5.3	SLAM in a corridor of length 22.7 meters	108
5.6	Conclusion	109
6	UWB/LiDAR Fusion SLAM via Step-by-step Iterative Optimization	111
6.1	Introduction	111
6.2	Problem Formulation	112
6.3	Localization Using UWB Measurements Only	113
6.3.1	EKF-based Range-Only Localization	113
6.3.2	PF-Based Robot's State Estimation	114
6.4	Scan Matching and Mapping	116
6.4.1	Fine-tuning the Robot's State Using Scan Matching	117
6.4.2	Mapping	120
6.4.3	State Correction	121
6.5	Experimental Results	122
6.5.1	Experimental Environment and Parameters Selection	122
6.5.2	Influence of Baseline on SLAM System	124
6.5.3	Proposed Fusion SLAM vs. Existing Methods	125

6.5.4	Accuracy of UWB Beacons' State Estimation vs. Method in Chapter 5	127
6.6	Conclusion	127
7	Autonomous Exploration Using UWB and LiDAR	131
7.1	Introduction	131
7.2	Technical Approach	133
7.2.1	Dual-UWB/LiDAR Fusion SLAM	133
7.2.2	Dual-UWB Robotic System	134
7.3	Global Path Planning	137
7.4	Where-To-Explore Path Selection	138
7.5	Collision-free Navigation	139
7.6	Experimental Results	140
7.6.1	Hardware Platform and Experimental Physical Environment	140
7.6.2	Proposed Autonomous Exploration vs. Manual Exploration	141
7.6.3	Proposed Autonomous Exploration System vs. An Existing Autonomous Exploration System	142
7.6.4	Illustration of an Exploration Process in the Garden	144
7.6.5	Ambiguous Boundaries	145
7.7	Conclusion	146
8	Conclusion and Future Research	147
8.1	Conclusions	147
8.2	Recommendations for Future Research	150
9	Appendix	153
9.1	Derivation of numerical optimization based weight solver	153
9.2	Derivation of model evaluation based weight solver	155
	Author's Publications	159
	Bibliography	161

List of Figures

1.1	The general framework of an autonomous robotic system.	2
1.2	An example of some challenge sequences among the currently popular datasets. (a) includes the challenge attributes of illumination variation, motion blur, in-plane rotation and scale variation. (b) includes the challenge attributes of occlusion, background clutters and in-plane rotation. (c) includes the challenge attributes of occlusion, background clutters and deformation.	4
1.3	Examples of tracking performance evaluation on single feature and the general scheme of ensemble tracking. The features 1-4 in (a) are Color Name (CN), Histogram of oriented gradients (HOG), <i>conv5-4</i> of VGG-19 and <i>conv4-4</i> of VGG-19, respectively.	10
1.4	A block digram of proposed system.	13
1.5	A block diagram of proposed fusion system. $\mathbf{p}_t^{\text{sc}}$ is the laser point cloud.	14
1.6	How UWB beacons define the to-be-explored region.	17
3.1	A framework of the proposed tracking algorithm. The correlation tracker is used to track the target efficiently. The tracker records the past predictions to discriminate the occlusion and model drift. The event-triggering model is used to produce triggering signals, labeled in different colors, to activate the corresponding subtasks, respectively. The detection model is used to re-detect the target when model drift happens.	41
3.2	Illustration for drift detection and restoration. The bounding boxes colored in green, red solid, red dash and yellow denote ground truth, ETT prediction, the prediction without re-detection and corrected results after detecting tracking failure, respectively.	49
3.3	Illustration of the event-triggered decision tree. The event-triggered decision model will produce one or multiple events based on the input from occlusion and drift identification model, which provides the information of short-term tracking state evaluation.	51
3.4	Illustration of the representation: (a) shows the composition of sampling-pool, which is divided into three portions, namely, support samples, high confidence samples and re-sampled samples; and (b) shows the updating methods for the first two parts in (a). . . .	55

3.5	Quantitative results on the benchmark datasets. The scores in the legends indicate the average Area-Under-Curve value for precision and success plots, respectively.	58
3.6	Quantitative results on OTB-100 [2] for 8 challenging attributes: motion blur, illumination variation, background clutters, occlusion, out-of-view, deformation, scale variation and out-of-plane rotation. .	61
3.7	Accuracy-robustness ranking plot for the state-of-the-art comparison.	64
4.1	A framework of the proposed trackers. A CF response map is generated using each single feature, which is then fed to the proposed weight solver to find proper weights. The evaluated weights are applied to estimate the target's position and update the CF models.	71
4.2	Accuracy of our approach with different values of ϵ and C	80
4.3	Quantitative results on the OTB-50 dataset with 51 videos. The scores reported in the legend of the left and right figures are precision score at 20px and average AUC values, respectively.	81
4.4	Quantitative results on the OTB-100 dataset with 100 videos. The scores reported in the legend of the left and right figures are precision scores at 20px and average AUC values, respectively.	81
4.5	Quantitative results on the TempleColor dataset with 129 videos. The scores reported in the legend of the left and right figures are precision scores at 20px and average AUC values, respectively. . . .	82
4.6	Expected overlap curves plots. The score reported in the legend is the EAO score which takes the mean of expected overlap between two purple dot vertical lines.	82
4.7	Expected overlap curves plots. The score reported in the legend is the EAO score which takes the mean of expected overlap between two purple dot vertical lines.	83
4.8	Quantitative results on OTB-100 for 8 challenging attributes: background clutters, deformation, fast motion, in-plane rotation, illumination variation, motion blur, occlusion and out-of-plane rotation. .	85
4.9	Precision and success plots on the LaSOT dataset with 280 videos. The scores reported in the legend of the left and right figures are precision scores at 20px and average AUC values, respectively. . . .	86
4.10	Qualitative tracking results with the proposed two trackers and other state-of-the-art trackers.	90
5.1	Illustration of two versions of the same relative geometry of four nodes: one is translated and rotated version of the other one. . . .	97
5.2	Our UWB/LiDAR SLAM system.	103
5.3	UWB/LiDAR fusion in four cases: a) no scan matching, no correction; b) with UWB/LiDAR scan matching where $\gamma = 0.65$, no correction; c) with UWB/LiDAR scan matching where $\gamma = 0.65$ and correction; d) with LiDAR-only scan matching where $\gamma = 10^{-6}$ and correction. The green "+" denotes the final position of the robot. .	104

5.4	Beacon(s) drops/joins/moves while SLAM is proceeding.	106
5.5	Comparison of our UWB/LiDAR-based SLAM with HectorSLAM [4] at different robot's speeds. To build the maps, the robot moves for one loop of the same trajectory as shown in Fig. 5.3.	107
5.6	Comparison of our UWB/LiDAR-based SLAM with HectorSLAM [4] in a corridor of length 22.7m.	108
6.1	An illustration of the proposed adaptive-trust-region scan matcher. The optimization starts at the robot's coarse state estimation and then a group of particles are sampled based on an adaptively learned proposal distribution. At each iteration, the optimal particle (i.e., the one with the largest weight) is found and set as the initial position for the next iteration. The red circle denotes the approximate size of search region.	118
6.2	Hardware platform and experimental physical environment.	123
6.3	Comparison of the proposed fusion SLAM with other SLAM approaches in workshop scenario.	129
6.4	Comparison of the proposed fusion SLAM with other SLAM approaches in garden scenario.	130
7.1	The framework of the proposed autonomous exploration system using UWB and LiDAR.	132
7.2	Exploration in two different scenarios: indoor and outdoor. Figures from left to right are the maps obtained from the manual exploration (left), the autonomous exploration (middle), and the heat map that shows the difference between these two exploration results (right). Dark green, dark purple and yellow colors in the built map represent unexplored region, explored region and occupied region, respectively. The blue line is the trajectory that the robot has travelled.	141
7.3	Comparison of the proposed method and the Multi-RRT exploration scheme [5] under two different scenarios: indoor and outdoor. Figures show the map built at different timestamps.	143
7.4	Exploration process in a garden scenario. The yellow circles represent the UWB beacons location estimate while the blue circle represents the robot's location estimate. The red line is the planned path to the selected UWB beacon.	144
7.5	Ambiguous boundaries when exploring in the garden.	145

List of Tables

3.1	Technical comparisons among similar state-of-art trackers.	42
3.2	The feature representation for the defined events.	50
3.3	Comparisons among baseline trackers.	59
3.4	A time consumption comparison. The mean overlap precision (OP) (%), distance precision (DP) (%) and mean FPS over all the 100 videos in the OTB-100 [2] are presented. The two best results are displayed in red and blue, respectively.	59
4.1	Tracking performance evaluation with different individual feature and their combination. Red : the best. Blue: the second best.	77
4.2	Evaluating the impact of adaptively adjusting the regularization penalty of CF. The scores listed in the table denote OP (@AUC) and DP (@20px), respectively.	77
4.3	Comparisons of our approach with the state-of-the-art trackers under distance precision (@20px) on OTB-100 and TempleColor. Red : the best. Blue: the second best.	79
4.4	Comparisons to the state-of-the-art trackers on the VOT2016. The results are presented in terms of EAO, accuracy and failure. Red : the best. Blue: the second best. Green : the third best.	84
4.5	Time evaluation on the steps of the proposed tracker.	90
5.1	Notations for the symbols used in this chapter.	94
5.2	Averaged errors/stds. of five UWB beacons' pose estimates.	105
6.1	Guideline for the parameter tuning, where $\uparrow$ indicates increasing in value while $\downarrow$ means decreasing in value.	124
6.2	Evaluation on the influence of different baseline.	126
6.3	Averaged errors/stds. of five UWB beacons' pose estimates.	127
7.1	Comparisons among the proposed exploration, manual exploration and Multi-RRT exploration. The * indicates that the exploration is not completed due to the accumulated errors. The + indicates that the result is an average over 20 independent experiments in the same environment.	142

Symbols and Acronyms

Symbols

A^T	Transpose of matrix A
A^{-1}	Inverse of matrix A
$\mathbb{R}$	Set of real numbers
$\mathbf{1}_n$	A column vector with n elements all being one
∇	The gradient operator
$\mathcal{R}^n$	The n -dimensional Euclidean space
$ \cdot $	The absolute value of a vector or matrix in Euclidean space
$\ \cdot\ $	The 2-norm of a vector or matrix in Euclidean space
$\odot$	The Hadamard (component-wise) product
$\otimes$	The Kronecker product
$\langle \cdot, \cdot \rangle$	The inner product of two vectors

Acronyms

SLAM	Simultaneously Localization and Mapping
RRT	Rapidly-exploring Random Tree
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
CF	Correlation Filter
CNN	Convolutional Neural Network
EOA	Expected Averaged Overlap
HOG	Histogram of Oriented Gradients
CN	Color Name
UWB	Ultra-wideband
NLOS	Non-Light-Of-Sight

EKF	Extended Kalman Filter
RBPF	Rao-Blackwellized Particle Filter
DWA	Dynamic Window Approach
SVM	Support Vector Machine
FPS	Frames Per Second
KCF	Kernelized Correlation Filter
SIFT	Scale-Invariant Feature Transform
KKT	Karush-Kuhn-Tucker
OP	Overlap Precision
DP	Distance Precision
a.k.a.	also known as

Chapter 1

Introduction

Industrial and technical applications of unmanned ground vehicle are continuously gaining importance in recent years, in particular under considerations of accessibility (inspection and exploration of sites that are dangerous or inaccessible to humans, such as hazardous environments), reliability (uninterrupted and reliable execution of monotonous tasks such as surveillance) or cost (transportation systems based on autonomous mobile robots can reduce labour cost). Due to fruitful results in the past decades, mobile robots are gradually entering our normal life and some of them have already been widely used in surveillance, inspection and transportation tasks [6].

However, the existing mobile robots can only execute limited tasks in the pre-determined environment due to various challenges from the real world, such as unexpected circumstances, unstable perception under various illumination conditions, manipulating small objects, *et al.* . Thus how to empower the mobile robot to carry out the task in ad-hoc and clutter environment, such as shopping malls [7], hospitals [8] and airports [9], is still an open topic.

Autonomous mobile robot is a complex system as it is a system integration of several challenges yet crucial modules. Figure 1.1 illustrates the overall framework of an autonomous robotic system, which can be generally categorized as follows.

FIGURE 1.1: The general framework of an autonomous robotic system.

1. *Simultaneously localization and mapping (SLAM)*. The goal is to build and update a coarse/fine map of an unknown environment while simultaneously keep tracking the robot's location within it, with the environmental information sensed from various sensors, such as LiDAR and camera.
2. *Navigation*. The task is to find a geometric feasible path from a given particular location to a given goal under some constraints, such as shortest path, minimum time, motion dynamical of the robot, collision-free with obstacles detected by the on-board sensors, *et al.* , and control the robot to move alongside the planned path without any collision.
3. *Robotic perception*. It empowers the robot to perceive, comprehend, and reason about the surrounding environment from sensory data, thus the robot can take an appropriate action and reaction in the real-world situations. Robotic perception is related to many applications in robotics where sensory data and artificial intelligence/machine learning techniques are involved. Examples of such applications are object detection, object tracking, human/pedestrian detection, activity recognition, motion prediction *et al.* .

Research on motion planning has been carried out in decades. Some sophisticated motion planning approaches are proposed and applied in the field of robotics, for examples, grid-based search (i.e. A* planner [10] and D* planner [11, 12]), reward-based approach (fuzzy Markov decisions processes [13]), the probabilistic roadmap planner [14], artificial potential fields [15, 16], the rapidly-exploring random tree (RRT) planner [17, 18], deep reinforcement learning based approach [19], *et al.* . On the other hand, even if the technologies related to SLAM and robotic perception have been deeply explored in recent years, some challenging issues still remain to be solved before the robot can carry out general tasks, such as working harmony with human being which requires an accurate object detection and tracking approach, executing the task in dynamic environments which requires a robust SLAM system. Therefore, this thesis focuses on the problems involving these two topics, named sensor fusion for SLAM and its one potential application (autonomous exploration), and visual object tracking.

1.1 Motivation and Objectives

1.1.1 Visual Object Tracking

Visual object tracking is one of the core research problems with a wide range of applications in robotics system, such as using unmanned ground vehicle (UGV) or unmanned aerial vehicle (UAV) to track a dynamic target (*e.g.* vehicle, bicycle, pedestrian) on the ground [20–23], UGV following human in the indoor environment [24–28], *et al.* . Recently, lots of research works have been carried out to integrate object tracking with SLAM to robustify the SLAM system in the dynamic scene (i.e., shopping mall) [29–32]. These methods are generally attempt to address the issue of moving obstacles in the SLAM problem. An advanced object tracking system can facilitate the data association among multiple moving objects, and then eliminate the influence from moving obstacles, thus robustify the SLAM

system in dynamic environment. The above mentioned applications motivate us to explore the visual object tracking task.

Despite substantial progress has been made in the past decades, there still remain some challenging issues, such as abrupt motion, illumination change and appearance variation. Figure 1.2 illustrates some examples of the challenge sequences among the recently popular datasets. According to its actual applications, visual object tracking can be categorized as long-term and short-term tracking.

(a) *MotorRolling* sequence in OTB-100 [2]

(b) *Fish* sequence in TempleColor [33]

(c) *Glove* sequence in VOT2016 [3]

FIGURE 1.2: An example of some challenge sequences among the currently popular datasets. (a) includes the challenge attributes of illumination variation, motion blur, in-plane rotation and scale variation. (b) includes the challenge attributes of occlusion, background clutters and in-plane rotation. (c) includes the challenge attributes of occlusion, background clutters and deformation.

1.1.1.1 Robust Visual Object Tracking via Event-Triggered Tracking Failure Restoration

Long-term visual object tracking is an important problem in computer vision community, with a wide range of applications such as automated surveillance, robot navigation and many others. Great progress has been made in recent years [34–41]. Yet numerous practical factors, such as heavy occlusion, appearance variation, illumination variation, abrupt motion, deformation and out-of-view, can easily lead to model drift and thus they need to be taken into account.

The long-term visual tracking is a complicated yet systematic task, which cannot be solved independently through either tracking or detection. Several interesting trackers are proposed recently to address the long-term tracking problem, for instance, in [42] and [43]. In [42] an on-line random forest detector is proposed and in [43] short-term and long-term target appearance memories are stored to estimate the object location in every frame, and correct the tracker if necessary. However, detection on each frame is a complex and time-consuming task as the detector needs to search a large area to find the target candidate with the highest response, while tracking usually predicts approximate location of the target with *priori* information and searches the target within a small neighboring region. In many applications, rapid visual tracking is essential to ensure required system performances. For instance, in a scenario where an autonomous robot is to follow a given target, the planned path based on the tracked target is used as a reference trajectory for controlling robot and should be available promptly for implementing control actions, which requires the target to be tracked real-time in various conditions. On the other hand, it should also be noted that detection occurring at each frame is redundant in many cases. Actually, there is little change between two consecutive frames when the time interval is small and in this case just using tracking model without detection can accurately track the target [34]. Thus it is unnecessary to carry out detection at each frame, since short-term tracking algorithm could accurately track the target in most scenarios.

Recently, [44] propose to activate object re-detection in case of model drift. However, the moment of activation directly relies on the comparison of the confidence of current prediction with a predefined threshold, which could lead to unnecessary triggering of its re-detection module in many cases and thus increase computational cost. Thus, how to identify the model drift accurately is a critical problem, as object detection and model updating at right time will significantly improve the tracking accuracy and also accelerate the tracking speed. In [45], an approach is proposed to identify the occurrence of occlusion for target patch, and if so, an optimal classifier is chosen from the classifier-pool in terms of entropy minimization. However, occlusion does not mean tracking failure in many scenarios and most short-term trackers are able to tolerate certain level of occlusion except for heavy or long-term partial occlusion. Furthermore, directly replacing current tracking model with the optimal classifiers trained beforehand increases the probability of tracking failure, since it loses the latest frame-to-frame translation information, which is important to short-term tracking. In this case, it is preferable to weaken the weight of noisy samples so as not to contaminate the tracking model, by learning samples discriminatively via excluding heavy occlusion/background samples and decreasing the learning rate for partial occlusion.

Motivated by the above discussions and inspired by the idea of event-triggered control proposed in control engineering [46, 47], an event-triggered tracking (ETT) framework with an effective occlusion and model drift identification approach by measuring the temporal and spatial tracking loss is presented in this thesis, in order to simultaneously ensure fast and robust tracking with the required accuracy.

With such a novel tracking framework, the tracker is able to obtain a more robust the short-term tracking performance and also properly address model drift problem for long-term visual tracking. Our proposed scheme is tested on a frequently used benchmark datasets OTB-100 [2]. The experimental results demonstrate that the proposed tracking scheme yields significant improvements over the state-of-the-art trackers under various evaluations conditions. More importantly, the proposed

tracker runs at real-time and is much faster than most of the online trackers such as those in [36, 39, 43, 44, 48, 49]. The detail elaboration is presented in Chapter 3.

1.1.1.2 Robust Visual Object Tracking via Multi-feature Reliability Re-determination

Visual object tracking is one of core research problem with a wide range of applications, such as vehicle tracking, surveillance, robotics. In recent years, correlation filter (CF) has been commonly integrated with deep features among the visual community, which has significantly improved the tracking accuracy and robustness among the most popular datasets, see [50, 51] for examples. Despite substantial progress has been made in the past decades, there are still some challenge issues remaining to be conquered, such as abrupt motion, illumination change, occlusion and appearance variation, which can easily result in the drift on the learned tracking model.

Generally, the CF-based visual tracking task can be accomplished by recursively predicting the target's location $\mathbf{B}_t$ and updating the discrimination model Ξ_t as follows

$$\Delta_t^* = \arg \max_{\Delta_t} \Omega(\mathbf{I}_t, \Xi_{t-1}), \quad (1.1)$$

$$\mathbf{B}_t = \mathbf{B}_{t-1} \oplus \Delta_t^*, \quad (1.2)$$

$$\Xi_t = \mathbf{U}(\mathbf{B}_t, \mathbf{I}_t, \Xi_{t-1}), \quad (1.3)$$

where $\mathbf{I}_t$ denotes the t -th image frame, $\Omega(\mathbf{I}_t, \Xi_{t-1})$ is a user specified model to predict the variation of the target's states (i.e. translation, rotation and scale) from $(t-1)$ -th frame to t -th frame, $\oplus$ is a state updating operator, $\mathbf{U}(\cdot)$ updates the discriminative model Ξ_t using the t -th image frame upon the previously learned Ξ_{t-1} . Therefore, any error resulted from the prediction step in (1.1) will be transferred to the target's state estimation in (1.2), and thus introduce noisy training samples when updating the discrimination model Ξ_t in (1.3). Therefore, such type

of errors will be gradually accumulated and finally contaminate the discriminative model Ξ_t , thus result in model drift.

To alleviate such model drift, various approaches have been proposed. They can be generally categorized as: 1) identifying noisy samples (i.e. occlusion, motion blur) in order to avert poor model updating see [44], 2) detecting tracking failure or model drift, and then reinitializing the tracking model when necessary by re-locating the target see [45, 52], and 3) learning a stronger discrimination model Ξ_t by applying advanced features (i.e. deep feature) see [50, 53]. However, both identifying the model drift and re-locating the target are not an easy task due to the lack of training samples. On the other hand, significant improvements achieved by deep learning based trackers on the popular datasets, such as OTB-100 [2] and TempleColor [33], are benefited from deep learning technologies, while such improvements are usually at the expense of computational cost.

Therefore, another objective of this thesis is to fuse multiple features to alleviate the issue of model drift. We attempt to alleviate the model drift by formulating a proper discrimination model which can integrate multiple weaker features and take advantage from each of them. Firstly, motivated by the method presented in [54] which integrates multiple weak features in a cascade model and thus constructs a stronger face detection model. Secondly, note that each feature usually has its advantage and also limitation on representing the target, for example, deep feature provides more semantic information about which kind of object it belongs to while handcraft feature presents more detail information about the relationship among pixels. Thus a single feature generally cannot fulfil various tracking tasks, for example, deep feature presents more semantic information about the category that the object belongs to, while handcraft feature usually focuses on the relationship among pixels. So the visual tracking behaviour upon different feature representation varies significantly, as illustrated in Fig. 1.3(a). Thirdly, inspired by the ensemble tracker proposed in [55] which selects a suitable tracker from a series of independent CF trackers as the tracking result of the current frame. Mathematically, this kind of decision-level fusion tracker can be generally formulated as to find

an appropriate evaluation function $f(\cdot)$ that treats the bounding boxes predicted from multiple weaker trackers as inputs, as illustrated in Fig. 1.3(b).

Based on the aforementioned discussion, we propose to formulate multiple types of feature in one discrimination model to yield a stronger yet more robust tracker. In detail, we propose a reliability re-determination correlation filter by maintaining a reliability score for each feature to online adjust the importance of them. Furthermore, we provide two different solutions to determine these weights: 1) numerical optimization which iteratively optimizes the reliability and CF model, 2) model evaluation which evaluates reliabilities by learning an extra discriminative model. In summary, what we focus on is how to build a reliability re-determination model to correctly present the importance of each feature. The experimental results have demonstrated that the proposed reliability re-determination scheme can effectively alleviate the model drift and thus robustify the tracker. Especially on VOT2016, the proposed two trackers in Chapter 4 achieve outstanding tracking results in terms of expected averaged overlap (EAO) (i.e. **0.453** and **0.428** respectively), which significantly outperforms the recently published top trackers. The detailed elaborations are presented in Chapter 4.

1.1.2 Sensor Fusion for SLAM

Simultaneous localization and mapping, a.k.a SLAM, has attracted immense attentions in the mobile robotic literature, and many approaches use laser range finders (LiDARs) due to their accurate range measurement to the nearby objects. There are two basic approaches to mapping with LiDARs: feature extraction and scan matching. The feature extraction based SLAM methods attempt to extract the important yet unique features, such as corners and lines from indoor (rectilinear) environments, and trees from outdoor environments, to assist the localization of robot and construct the map. While the scan matching based approaches match point clouds directly and locate the robot's poses via map constraints, which are much more adaptable than feature extraction approaches as they depend less on

(a) Tracking results obtained from different features.

(b) Ensemble tracking scheme.

FIGURE 1.3: Examples of tracking performance evaluation on single feature and the general scheme of ensemble tracking. The features 1-4 in (a) are Color Name (CN), Histogram of oriented gradients (HOG), *conv5-4* of VGG-19 and *conv4-4* of VGG-19, respectively.

environment. Among all existing scan matching algorithms, GMapping [56] and HectorSLAM [4] are arguably the most well-known and widely used algorithms. GMapping needs odometry input while HectorSLAM is an odometry-less approach. Apart from their respective advantages, one common drawback is that they are very vulnerable to the accumulated errors. These errors may come from long run operation of odometry such as GMapping which directly extracts odometry information from odometry sensor. The accumulated errors may also come from the SLAM algorithm itself such as HectorSLAM where the error of robot's pose at current time step will be passed via scan matching procedure to the grid-map, which will in turn impair the estimation of robot's pose at next time step.

Generally, when using one sensor alone, the robot's location $\mathbf{p}_t$ and the map $\mathbf{M}_{\rightarrow t}$ built up to time t can be updated recursively over time as follows.

$$\Delta_t^* = \arg \min_{\Delta_t} \mathcal{D}(M(\mathbf{p}_{t-1,r} + \Delta_t, \mathbf{p}_t^{\text{snr}}), \mathbf{M}_{\rightarrow t-1}), \quad (1.4)$$

$$\mathbf{p}_{t,r} = \mathbf{p}_{t-1,r} + \Delta_t^*, \quad (1.5)$$

$$\mathbf{M}_{\rightarrow t} = M(\mathbf{p}_{t,r}, \mathbf{p}_t^{\text{snr}}) \cup \mathbf{M}_{\rightarrow t-1}, \quad (1.6)$$

where $\mathbf{p}_{t,r}$ denotes the robot's location at time t , $\mathbf{p}^{\text{snr}}$ denotes the sensor's measurements at time t , e.g., the scan endpoints in 2D LiDAR, the point cloud in 3D LiDAR or the camera-captured images, Δ_t denotes the robot's displacement from time $t-1$ to time t , $M(\cdot)$ is a function that returns a map built upon $\mathbf{p}_t^{\text{snr}}$ while the robot is locating at $\mathbf{p}_{t-1,r} + \Delta_t$, $\mathcal{D}(\mathbf{A}, \mathbf{B})$ is a user-defined distance metric that measures the difference between $\mathbf{A}$ and $\mathbf{B}$, and $\mathbf{M}_{\rightarrow t}$ denotes the map learnt up to time t , which is obtained by merging the map $M(\mathbf{p}_t^{\text{snr}}, \mathbf{p}_{t,r})$ observed at time t and the map $\mathbf{M}_{\rightarrow t-1}$ learnt up to time $t-1$.

Any error that occurs in the robot's displacement estimate Δ_t in (1.4) is retained in the robot's location estimate in (1.5), and is thus retained in the learnt map $\mathbf{M}_{\rightarrow t}$ in (1.6). Therefore, the error is accumulated over time. One way to eliminate the errors accumulated over time is to let the robot revisit the areas that have already been explored in order to generate loop closures [57]. However, this operation would require more computational resource. One another way to eliminate accumulated errors and enhance the robustness in LiDAR-based SLAM is by sensor fusion [58]. In this thesis, we present the work of fusing LiDAR sensor with ultra-wideband (UWB) sensors to eliminate such accumulated error. The objective of fusing UWB and LiDAR is to 1) provide not just landmarks/beacons but also a detailed mapping information about surrounding environment; 2) improve the accuracy and robustness of UWB-based localization and mapping.

Why do we choose UWB? In many practical environments such as enclosed areas, urban canyons, high-accuracy localization becomes a very challenging problem in the presence of multipaths and non-light-of-sight (NLOS) propagations. Among

various wireless technologies such as the Bluetooth, the Wi-Fi, the UWB and the ZigBee, UWB is the most promising technology to combat multipaths in cluttered environment. The ultra-wide bandwidth in UWB results in well separated direct path from the multi-paths, thus enabling more accurate ranging using time-of-arrival of the direct path.

However, the fusion is hindered by the discrepancy between the accuracy of the UWB mapping and that of LiDAR mapping. UWB has lower range resolution than LiDAR (the laser ranging error is about 1cm which is about one-tenth of UWB ranging error) so that UWB cannot represent an environment in the same quality as LiDAR can do. In this case, fusion by building LiDAR map directly on top of UWB localization results is not a proper solution.

Meanwhile, in this thesis, two different fusion schemes are proposed to address such a fusion issue. Their core idea is similar and can be summarized as three steps: 1) UWB-only SLAM: coarsely estimate the states of robot as well as UWB beacons using UWB ranging measurements. 2) Scan matching: refine the robot's pose using LiDAR scanning or both of LiDAR and UWB ranges. 3) Correction: correct the UWB beacons' state with the refined states of robot.

For step 2, we consider to refine the robot's coarse pose obtained from UWB ranging measurements by feeding it to a scan matching procedure. However, based on using LiDAR and UWB ranges or LiDAR ranges alone to refine the robot's state, two different fusion schemes are presented in this thesis, named "one-step" fusion scheme and "step-by-step" fusion scheme, respectively. The significant difference between these two fusion schemes is that the proposed methodology for handling robot's states refinement is different.

A block diagram of one-step fusion scheme is shown in Fig. 1.4, where the system collects all peer-to-peer UWB ranging measurements, including robot-to-beacon ranges and beacon-to-beacon ranges at time t , based on which the robot and beacons' 2D positions and 2D velocities are estimated using extended Kalman filter (EKF). Then the system feeds the state estimation as well as the observations of

UWB and LiDAR ranges to scan matching procedure in order to update map and find the optimal state offset.

FIGURE 1.4: A block digram of proposed system.

We note that the one-step fusion scheme has some limitations and thus the second fusion scheme is proposed to address them as illustrated in Fig. 1.5 , which has four fundamental differences from it:

1. While Fig. 1.4 uses an EKF to estimate all UWB sensors' locations and bearings, we propose to use Rao-Blackwellized particle filter (RBPF) together with a dual-UWB setup so that the robot's bearing can be estimated more smoothly which is crucial in the subsequent mapping.
2. In the scan-matching optimization, while Fig. 1.4 uses a composite loss, i.e., sum of UWB loss and LiDAR loss, where the trade-off between them needs to be manually tuned, we propose a fundamentally different optimization procedure that iteratively fine-tunes the robot's states.
3. Unlike Fig. 1.4, the proposed scan matching method, which is based on adaptive-trust-region RBPF, does not linearize the objective function defined in Fig. 1.4 thus reducing the chance of being trapped into a local minimal.
4. Fig. 1.4 includes the UWB beacons' states in the scan-matching optimization, however, we note that it is unnecessary as the LiDAR measurements have

no direct impact on the UWB beacons. Hence, we propose to rectify the beacons' states only after the scan matching is done.

The detail elaborations involving these two UWB/LiDAR fusion schemes are presented in Chapter 5 and Chapter 6, respectively.

FIGURE 1.5: A block diagram of proposed fusion system. $\mathbf{p}_t^{\text{sc}}$ is the laser point cloud.

1.1.3 UWB/LiDAR Fusion for Autonomous Exploration

Unknown environment exploration is a popular application of SLAM and it is among the most important topics in mobile robot research field, which usually proceeds in a successive manner, i.e., merging the newly explored region to the existing (built) map, by using local sensor information. The objective of the exploration might be divergent, such as mapping for the unknown environment, searching for

the victims after some accidents [59], military assignments, *et al.* . For such applications, an exploration has to be carried out very efficiently, i.e., with the least possible amount of time to construct a map while maintain the map's quality.

So far many efforts have been devoted to improving the efficiency and robustness of autonomous exploration. For example, [60] utilizes flying robots to build a coarse map to assist the ground robot to find a collision-free path; [61–63] focus on optimizing the robot's navigation path so that the environment can be explored more efficiently; [64] and [65] present efficient exploration schemes with introduction of deep learning strategies. Depending on whether the robot is operated by a human or not, the exploration can be categorized into manual exploration or autonomous exploration. The former is usually regarded as a SLAM problem while the latter needs to solve all problems as mentioned above. Autonomous exploration has many potential applications especially in ad-hoc scenarios.

To perform environment exploration autonomously, the robotic system not only addresses the SLAM problem discussed in Section 1.1.2, but also determines the intermediate points, which define the robot's exploration behaviour, and navigate the robot itself from the current point to a given destination.

The where-to-explore point is an intermediate point about where the robot should move to further extend its map. The choice of these points is an essential module of autonomous exploration, since it determines the robot's exploration behaviour and thus influences the accuracy and efficiency of the exploration process. Ideally, choosing where-to-explore points should take into consideration of gaining more new information of the environment and existence of a feasible path to them. But how to meet these conditions and how to define the "gaining more new information" are the core problems that will be addressed in this thesis.

In the past decades, several where-to-explore point choosing strategies have been proposed, categorized as human directed, choosing based on frontier, and choosing based on the additional background information. Human directed exploration focuses on the map construction and the interaction between human being and

the robots, while where to explore is determined by the person, thus this kind of exploration can usually achieve a satisfactory result. Frontier-based approaches [5, 66, 67] are used to automate the exploration. Frontiers are the regions on the boundary between open space and unexplored space. By moving to new frontiers, a mobile robot can extend its map into new territories until the entire environment is explored. One drawback of this approach is that the computational cost of frontier edge extraction increases rapidly as the explored map is expanded. To handle this concern, some works explore to speed up the detection of frontier edges [5, 68]. Another drawback is that the frontier points detected in the cluttered region are usually noisy which may provide inaccurate guidance for exploration.

Another way is to use additional knowledge (if any) about the unknown environment to facilitate the choosing of where-to-explore points. Such information includes a rough layout of the environment [69], semantic information [70] and environmental segmentation [71]. However, preparing such information is usually a time-consuming task.

In the proposed exploration system, we place in the to-be-explored region multiple UWB beacons whose locations are all unknown and need to be estimated while the exploration is on-going. The convex hull of the UWB beacon positions defines a region about where the robot is going to explore. An example is shown in Fig. 1.6. We also consider UWB beacons as the robot's where-to-explore points. For example, the robot can pick one UWB beacon, to which the path is less explored, as its next exploration point to move to. The extent to which the path has been explored can be measured from the mapping process.

Then, a collision-free navigation system, the executive module of the robot is implemented to navigate the robot. It usually consists of motion planning and motor control. Generally, the robot will receive a command to indicate its next destination from the decision making centre (a.k.a. the brain of the robot), then a global path to such destination will be found based on a global path planner, such as A* planner [10], D* planner [11, 12], RRT planner [17, 18], *et al.* . With the

FIGURE 1.6: How UWB beacons define the to-be-explored region.

planned path, the robot can infer the control command to drive the robot at each timestamp according to control algorithms, such as trajectory following [72, 73]. However, there might be some unexpected obstacles locating at the surrounding of the planned path. In this case, the robot has to take certain actions, such as stop itself or re-find a new feasible path to bypass the obstacles. Usually, the robot does not need to re-find the path to the final destination, since it is a time consumption process. Instead, the robot only needs to find a path among a small region which could enable the robot to bypass the obstacles by using some typically local motion planning approaches, such as dynamic window approach (DWA) [74, 75], velocity obstacles [76, 77], time-to-collision approaches [78–80], *et al.* .

Based on the aforementioned discussion and noting that the fusion SLAM system proposed in Section 1.1.2 cannot estimate the bearing of robot correctly when the robot is not moving as only one UWB node is mounted on the robot, we build an autonomous exploration system which includes the following three aspects.

- Build a dual-UWB SLAM system based on the fusion scheme in Section 1.1.2 but with two non-trivial improvements over it: 1) A dual-UWB robot is built to improve the robustness of the robot's heading estimation; 2) The exploration is automated.
- Instead of moving to frontiers, our exploration system infers the abstract background of the environment based on UWB network. The robot in our system selects a UWB beacon to move until it arrives in the beacon's vicinity and the selection depends on how well the potential route is explored (less explored route is favoured), and the map is constructed while the robot is navigating itself among the environment with a dual-UWB/LiDAR SLAM system.
- We implement our global path planner according to RRT and local motion planner according to DWA, to navigate the robot collision-freely among the unknown environment.

1.2 Major Contributions

The main contributions of the thesis are summarized as follows:

- **Event-triggered tracking for long-term object tracking.** A systematic ETT framework is built by decomposing the visual tracker into multiple event-triggered modules which work independently and are driven by particular events. It also allows future upgrade so as to introduce more sophisticated models. Among the proposed tracking framework, 1) an occlusion and drift detection method is built by considering the temporal and spatial tracking loss. It enables the event-triggered decision model to accurately evaluate the short-term tracking performance, and to trigger the relevant module as needed. 2) A bias alleviation sampling approach for support vector machine (SVM) model is constructed. A sampling-pool with support samples, high

confidence samples, and re-sampling samples, is constructed to alleviate the influence of noisy or mislabeled samples. 3) Weighted learning for short-term tracking model is applied, which will learn partial occlusions weakly and reject the heavy occlusion samples completely. Extensively experiments on the popular dataset demonstrate the effectiveness of the develop event-triggered tracking. Such contributions are presented in Chapter 3.

- **Adaptive multi-feature reliability re-determination correlation filter tracking.** 1) We formulate a reliability re-determination correlation filter which considers the importance of each feature when optimizes the CF model, thus enables the tracker adaptively select the feature which is suit for the current tracking scenario. 2) Two different solutions, named numerical optimization and model evaluation, are proposed to online re-determine the reliability for each feature. 3) Two independent trackers are implemented based on the proposed two different optimization solutions for the reliability of each feature. Extensive experimental tests have been designed to validate the performance of the proposed two trackers on five large datasets, including OTB-50[1], OTB-100[2], TempleColor[33], VOT2016[3] and VOT2018[81].

The experimental results has demonstrated that the proposed reliability re-determination scheme can effectively alleviate the model drift. Especially on VOT2016, both of the proposed two trackers have achieved an outstanding tracking result in terms of EAO score, which significantly outperform the recently published top trackers, as shown in Chapter 4.

- **UWB/LiDAR Fusion For Cooperative Range-Only SLAM.** An UWB/LiDAR fusion SLAM framework is built to eliminate the issue of accumulated error in LiDAR/Camera only SLAM. In the proposed SLAM framework, no prior knowledge about the robot's initial position is required. Meanwhile, we have proposed two significantly different approaches to handle the scan matching which is the core module of fusing UWB and LiDAR. The first approach formulates a composite loss, i.e. sum of UWB loss and LiDAR loss. Then optimization is achieved by feeding UWB and LiDAR ranges as

well as the coarsely estimated states of robot and UWB beacons to the scan matching module. The second approach aims to overcome the drawbacks of the first solution, i.e. linearizing the composite objective function. An adaptive-trust-region RBPF is proposed to iteratively fine-tune the robot's states using coarsely estimated states of robot and LiDAR ranges. Overall, both of these fusion methods allow that UWB beacon(s) can be moved and number of beacon(s) may be varied while SLAM is proceeding. More importantly, the robot can move fast without accumulating errors in constructing map under the proposed fusion framework, and the map can be built even in feature-less environment, such as the corridor. These developed SLAM methods have been evaluated on two real world sites, indoor workshop and a long corridor, see Chapter 5 and Chapter 6 for details. The experiments are filmed and the video is available online ¹.

- **Autonomous Exploration Using UWB and LiDAR.** a) For the first time (to the best of our knowledge), UWB and LiDAR are fused for autonomous exploration. b) UWB beacons are applied in this developed exploration approach to cover the region-of-interest where the locations of beacons are estimated on the fly. c) UWB beacons are considered as robot's intermediate stops and a where-to-explore scheme is proposed for robot to select the next beacon to move to. d) The exploration can be done rapidly and can relieve the issue of error accumulation. e) The system integrates a motion plan module to enable collision-free exploration. The developed autonomous exploration system is validated in two real world scenarios, indoor cluttered workshop and outdoor spacial garden, as presented in Chapter 7. The experiments are also filmed and the video is available online ^{2,3}.

¹SLAM in workshop: <https://youtu.be/yZIK37ykTGI>

²Autonomous exploration in workshop: https://youtu.be/depGuH_h2AM

³Autonomous exploration in garden: <https://youtu.be/FQQBuIuid2s>

1.3 Organization of the Thesis

The remainder of this thesis is organized as follows:

- Chapter 2 presents a literature review on the techniques and algorithms involved in long-term and short-term visual object tracking, sensor fusion for SLAM and autonomous exploration.
- Chapter 3 describes the developed event-triggered tracking framework in detail, includes the core idea and motivation on the designation of each sub-modules. Extensive experiments on large benchmark datasets are designed to validate the effectiveness and efficiency of the proposed method, especially on handling the issue of model drift in the visual tracking.
- Chapter 4 presents the developed adaptive multi-feature reliability re-determination correlation filter and its application on visual object tracking. Meanwhile, two different solutions on dynamically finding the optimal reliability score for each feature are discussed detailedly. Extensive experimental results on five large datasets are given which can validate the effectiveness of the proposed method.
- Chapter 5 introduces an UWB/LiDAR fusion framework to eliminate the accumulated error in SLAM. Experiments on real world and discussion on the effectiveness of handling accumulated error are given in this chapter.
- Chapter 6 discusses the drawbacks of the fusion SLAM approach presented in Chapter 5 and presents a solution to overcome such drawbacks. Meanwhile, comparison experimental tests on two real world scenarios are given to validate that the new proposed fusion scheme is superior than the one in Chapter 5.

- Chapter 7 presents an autonomous exploration system under the work done on the UWB/LiDAR fusion SLAM. Experiments are conducted in two different environments: a cluttered workshop and a spacious garden to verify the effectiveness of the proposed strategy.
- In Chapter 8, concluding remarks of this thesis and discussing on future works are given.

Chapter 2

Literature Review

2.1 Visual Object Tracking

Visual tracking has been studied extensively with numerous applications [82]. In this section, prominent visual object tracking approaches published in literature are introduced and reviewed.

2.1.1 Tracking-by-detection

The tracking problem in some cases are treated as a tracking-by-detection problem, which keep detecting the target at each tracking frame. Generally, a binary classification model is learned online/offline to find a decision boundary which has highest similarity with the given target. Lots of attentions were paid to learn a more discriminative model with less ambiguity to increase the tolerance on noisy training samples, thus improve the tracking accuracy, such as, multiple instance learning [83], SVM [39] and P-N learning [42] for examples. Kalal *et al.* [42] decompose the tracking task into tracking, learning, and detection, where the detection model is online trained from a random forest method. The tracking and detection facilitates each other, i.e., the training data for the detector is sampled according to the tracking result, and the detector re-initializes the tracker when it fails. Hare

et al. [39] propose to learn a joint structured output SVM to predict the object location, which avoided the need for an intermediate classification step.

However, the tracking efficiency is a possible issue for the trackers based on tracking-by-detection strategy, due to plenty of candidate samples needed to be classified at each tracking frame, since the tracking speed is significantly important for some applications. For example, [42] with efficiently trained random forest model can achieve about 25 frames per second (FPS), while [39] with explicit SVM model can only perform 13 and 5 FPS, respectively. The advantage of this tracking framework is the ability of long-term tracking due to the existing of the detection module which can recovery the tracker from some failure cases.

2.1.2 Correlation Filter Based Tracking

Recently, correlation filters (CFs) have been widely used in visual tracking [34–36, 38, 44]. The correlation filter-based trackers can train a discriminative correlation filter efficiently based on the property of the circulant matrix which transforms the correlation operation in spatial domain to an element-wise product in the frequency domain and thus tremendously reduces the tracking computational cost. Bolme *et al.* [35] initially apply correlation filter to visual tracking by minimizing the total squared error on a set of gray-scale patches. Henriques *et al.* [84] improve the performance by exploiting the circulant structure of adjacent image patches to train the correlation filter. Further improvement is achieved by proposing kernelized correlation filters (KCF) using kernel-based training and HOG features [34]. However, the above mentioned CFs only focus on predicting the target’s translation, but are not sensitive to scale variation. To handle such limitation, Danelljan *et al.* [38] propose an adaptive multi-scale CF to cover the target’s scale change. The above mentioned CFs are usually served as the baseline of CF-based trackers, which push forward the research on visual object tracking. However, they are still sensitive to some environmental variations, such as abrupt motion blur, illumination variation, occlusion, *et al.* .

To enhance the discriminative of the learned CF model, more sophisticated CFs are formulated by considering some new characteristics when training CF model [49, 85–88]. Such as, Tang *et al.* [85] derive a multi-kernel correlation filter which takes the advantage of the invariance-discriminative power spectrum of various features. Danellijan *et al.* [49] apply a spatial weight on the regularization term to address the boundary effects, thus greatly enlarge the searching region and improve the tracking performance. More improvements have been achieved by applying CNN features in [89]. Further more, [53] proposes a new spatial-temporary regularized correlation filter to further improve the accuracy and efficiency of [49], and [90] formulates an adaptive spatially-regularized correlation filters to improve [49] by simultaneously optimizing the filter coefficients and the spatial regularization weight. Liu *et al.* [86] reformulate the CF tracker as a multiple sub-parts based tracking problem, and exploited circular shifts of all parts for their motion modelling to preserve target structure. Mueller *et al.* [87] reformulate the original optimization problem by incorporating the global context within CF trackers. Sui *et al.* [88] propose to enhance the robustness of the CF tracker by adding a ℓ_1 norm regularization item to the original optimization problem, and an approximate solution for ℓ_1 norm is given. Tang *et al.* [85] derive a multi-kernel correlation filter by taking advantage of the invariance-discriminative power spectrums of various features to improve the performance. And further improvement on the tracking speed of [85] is presented in [91]. Galoogahi *et al.* [92] learn the correlation filter by taking the background information into consideration and achieve a satisfactory tracking speed. Lukezic *et al.* [93] introduce the channel and spatial reliability concepts to CF-based tracking and provide a learning approach which efficiently and seamlessly integrates such concept into the filter update and the tracking process.

Overall, it is obviously that correlation filter based tracking approach has been widely explored and also been improved from various aspects. Even through the experimental results show that CF-based trackers are competitive, none of them can generally perform well along various datasets, such as OTB-100[2], TempleColor[33] and VOT datasets[3, 81]. Therefore, a further research on CF-based tracking is

still necessary.

2.1.3 Deep Learning Based Tracking

Deep learning for visual object tracking has been widely explored and achieves favourable performance [51, 55, 94–96], due to its strong feature extraction ability. Some of them attempt to take advantage of the strong feature representation ability of CNN and reformulate the existing tracking framework (i.e. correlation filter) to enable the integration of CNN smoothly. For example, Wang *et al.* [94] propose a sequential tracking method using CNN features for visual tracking, which utilizes an ensemble strategy to avoid the network over-fitting. Ma *et al.* [97] employ multiple convolutional layers in a hierarchical ensemble of multiple independent discriminative CF based trackers. Danelljan *et al.* [89] apply the first convolutional layer of a CNN as the feature and then feed it into a CF-based tracking framework. Later, [89] is further improved by learning the CF with the continuous convolution operators in [98], and by proposing a factorized convolution operator and a compact generative model in [50]. Nam *et al.* [95] propose multi-domain CNN as the feature to represent the target and candidate samples. And its further real-time version is developed in [99].

Recently, Siamese neural network has attracted lots of attention among the visual tracking community. Tao *et al.* [100] initially introduce the Siamese neural network to fulfill the visual tracking task, which propose to train a discriminative model offline and then use it to evaluate the similarity between the target and candidate patches during tracking process. In [101], a positive sample generation network is built to generate positive samples for Siamese network in order to increase the diversity of the training data. Bertinetto *et al.* [102] train a fully-convolutional Siamese networks for visual object tracking, where the output of the network is a score map indicated the similarity between the target and the candidate, which greatly improves the tracking speed compared to [100] due to the avoidance of densely

sliding-window evaluation. Guo *et al.* [103] consider the Siamese network as a feature extracting approach and integrate it with a CF model for visual tracking. He *et al.* [104] propose a twofold Siamese network which takes the high-level semantic and low-level appearance information into consideration, Wang *et al.* [105] present a residual attentional Siamese network which formulates the CF within a Siamese tracking framework. Further, a Siamese region proposal network [106] and its improved one [51], a triplet loss Siamese network [107], a structured Siamese network [108], distractor-aware Siamese network [109] are proposed in the literature.

Meanwhile, there are some works which attempt to reduce the computational cost of deep learning based tracker. Valmadre *et al.* [110] tightly combine the CF with a CNN by interpreting the CF learner as a differentiable layer. Choi *et al.* [111] speed up the tracker by using multiple auto-encoders to compress the deep feature. Held *et al.* [112] offline learn a generic relationship between object motion and appearance from large number of videos and treat the online tracking as a testing process. However, the improvement on tracking speed is usually under the sacrifice of tracking accuracy.

Generally, the deep learning based tracker can either achieve a favourable tracking accuracy on many popular datasets such as VOT2016 [3], OTB-100 [2], while with an extremely low FPS, or achieve a real-time tracking speed but with the reduce on tracking accuracy. Moreover, the hardware requirement and high power consumption greatly fade its attraction, especially in the application of limited power supply, such as robotic systems.

2.1.4 Trackers with Model Drift Alleviation

The tracking failure is usually caused by the model drift/pollution in the presence of noisy samples. To alleviate such issue, various approaches have been proposed and they are generally categorized as: 1) detect the noisy samples (i.e., occlusion, motion blur, illumination variance) in order to avert poor model updating see [44],

2) detect tracking failure and reinitialize the tracking model see [45, 52] for examples, and 3) learn a stronger discrimination model Ξ_t and integrate it with stronger features (i.e. deep feature) see [50, 53] for examples. Such as, Zhang *et al.* [113] propose a discriminative feature selection method which couples the classifier score with the sample importance to enhance the robustness for visual tracking. Dong *et al.* [45] propose a classifier-pool to identify whether the current tracking state is occlusion or not, and an appropriate tracking strategy is proposed for each different tracking states, see occlusion and normal tracking. Yang *et al.* [114] propose an occlusion sensitive tracking approach via orthogonalizing templates from previous frames and removing their correlation, which also decompose the residual term into two components in observation model to take occlusion cases into consideration. Yu *et al.* [115] divide the multi-object tracking task into four processes (i.e. active, inactive, tracked and lost processes, respectively). A Markov Decision Process is then introduced to estimate the state transition.

However, accurately identifying noisy samples and tracking failure are not easy tasks, since the appearance of target may be variant at different frames and lack of the training samples, thus the false alarm is inevitable which will result in missing the learning of important frames.

2.1.5 Ensemble Tracking

Ensemble tracking is generally considered as a framework of integrating several independent modules (i.e. short-term tracking, object detection) to yield a robust tracker. Such as, Ma *et al.* [44] use a discriminative correlation filter to estimate the confidence of current tracking process to detect tracking failures and online learn a random forest classifier to re-locate the target. Further improvement is achieved by recording the past appearance of the target to restore the tracker when tracking failure occurs [116]. Hong *et al.* [43] introduces a biologic-based model to maintain a short-term and long-term memory of Scale-invariant feature transform (SIFT) key-points to detect the target and thus rectify the short-term tracker. Zhang *et*

al. [48] propose to learn a multi-expert restoration scheme where each expert is constituted with its historical snapshot, thus the best expert is selected to locate the target based on a minimum entropy criterion. Bertinetto *et al.* [36] propose to learn an independent ridge-regression model by taking consideration of colour cue to complement the traditional correlation filters. Zhang *et al.* [117] formulate a multi-task correlation particle filter for visual tracking where each particle is considered as an independent experts and the final tracking result is a weighted combination.

Others attempt to generate a suitable tracking output from a series of independent/weaker experts. For example, Wang *et al.* [118] propose a factorial hidden Markov model framework to jointly learn the unknown trajectory of the target and evaluate the reliability of each sub-tracker. Li *et al.* [119] propose a multi-expert framework which is built with the current tracker and the past snapshots of tracking model, then apply the unary and binary compatibility graph score to select a proper expert for tracking. Lee *et al.* [120] present a forward and backward trajectories analysis method to evaluate the sub-trackers from a tracker pool with variant feature extraction approaches. Wang *et al.* [55] consider each feature as an independent CF tracker, and built a selective model to find the suitable sub-tracker at each frame.

Overall, the ensemble tracking has achieved a great improvement, especially when combining handcraft features (i.e. HOG, CN) with CNN features (i.e. VGG, ResNet). Intuitively, this combination is meaningful since different features focus on different points, such as handcraft features capture the spatial details while CNN features provide more semantic information. Therefore, they can complement with each other and thus robustify the tracker. However, there are still some weak points among the existing ensemble tracking, for example, the final tracking output from [48, 55, 120] will potentially be dominated by an incorrect expert and drift the tracking model.

In this thesis, the tracker proposed in Chapter 4 follows the core idea of ensemble

tracking, i.e. attempting to integrate multiples types of feature to yield a stronger tracker. Different from the above mentioned methods, our proposed ensemble tracking framework focuses on how to re-determine the reliability for each feature. To validate the proposed ensemble tracking framework, two different solutions are proposed to adaptively learn such reliability score.

2.2 Simultaneous Localization and Mapping (SLAM)

The SLAM problem consists of two sub-problems, named mapping and localization, respectively. Both of them have many potential applications, such as autonomous driving, hazard environment exploration, *et al.* In the past decades, extensive researches involving SLAM have been taken under various sensors, such as LiDAR, Radar, UWB, Camera, Bluetooth, Wi-Fi, *et al.* . In the following paragraphs, a literature review related to SLAM is elaborated.

2.2.0.1 Wireless Sensor based SLAM

[121–128] propose to simultaneously localize robot(s) and static beacons based on robot-to-beacon ranging measurements and control input, i.e. odometry, IMU. Among these works, Tobias *et al.* [123] build a UWB Radar, which is consisted of two RX antennas and one TX antenna, to detect features around its vicinity for navigation. Christian *et al.* [128] use UWB technology to measure ranges and thus estimate the localization of both anchor (i.e. fixed UWB node) and user (i.e. moving tag), then it is implemented with IMU to predict the motion states for pedestrians. Joseph *et al.* [121] utilize the beacon-to-beacon ranging and robot-to-beacon ranging to build a UWB sensor network and thus locate the UWB beacons' position as well as robots. [121] is further improved in [129] by handling the cases of NLOS measurements and poor initialization. Blanco *et al.* [122] present a paradigm of Bayesian estimation with a RBPF to deal with the issue of time delay when adding the newly found beacons into SLAM system. Similarly, Emanuele *et*

al. [130] estimate the states of robots and beacons according to another ranging sensor, Received Signal Strength Indicator.

Generally, the wireless sensor based SLAM can provide an efficient mapping in the ad-hoc environment. Instead of constructing a detail map (i.e. including the obstacle information) involving the environment, these methods can only provide an abstract map about, i.e. how big the environment is or the relative positions among each nodes. Meanwhile, the wireless sensor more or less has some limitations, such as they are usually sensitive to the occlusion. This thesis further extends the paradigm in [122] by integrating the robot-to-obstacle ranges obtained from laser range finder (i.e. LiDAR). The new paradigm not only allows the SLAM system to localize the robot(s) and map the landmarks (i.e. the beacons), but also map the obstacles around the robot. Moreover, as the robot's pose can be estimated based on UWB ranging measurements, the heading information can be derived from its estimated trajectory thus we need no control input (i.e. encoders).

2.2.0.2 LiDAR based SLAM

Well-known approaches such as HectorSLAM[4], GMapping[56] have pushed forward the research on LiDAR-based SLAM. Gmapping proposes an adaptive approach to learn grid maps using RBPF while the number of required particles in RBPF can be dramatically reduced. HectorSLAM proposes a fast online learning of occupancy grid maps using a fast approximation of map gradients and a multi-resolution grid. Moreover, GMapping needs odometry input whereas HectorSLAM purely relies on the laser ranger finder. However, both HectorSLAM and Gmapping have a big issue of accumulated error due to lack of global reference, which greatly limit the size of working space. Later, the loop closure detection is introduced to deal with the accumulated error, such as, Granström *et al.* [131] propose to transform the 2D laser scans into pose histogram representations and match at this feature level. Himstedt *et al.* [132] present a machine learning based loop closure detection method by feeding the geometric and statistical features of point

cloud into a non-linear classifier model. Recently, Hess *et al.* [133] propose to use a branch-and-bound approach to efficiently detect the loop closure computing scan-to-submap matches. Even though the method proposed in [133] has achieved a real-time processing, but with highly computation cost. Overall, LiDAR-only based SLAM has the issue of the balance between accuracy and computational cost, which somewhat limits the application in some power limited cases. In this thesis, we consider to integrate the UWB sensor with LiDAR to handle the issue of accumulated error as well as the computational cost.

2.2.0.3 Vision based SLAM

Visual based SLAM systems are generally categorised into sparse SLAM [134–136] and dense or semi-dense SLAM [137–140]. For the former category, it usually represents the environment with a sparse set of features and thereby allows joint probabilistic inference of structure and motion. There are two main groups of approaches to jointly optimize the model and the robot’s trajectory, named filtering-based methods [134, 141, 142] and batch adjustment [135, 136, 143]. Montemerlo *et al.* [134] propose a FastSLAM which recursively estimates the full posterior distribution of robot pose and landmark locations. However, the accuracy of filtering based method is not so satisfactory while the computational cost for batch optimization is expensive, thus some researchers propose to make the visual odometry and trajectory optimization be carried out parallelly. Klein *et al.* [135] propose to split camera tracking and mapping in parallel threads to reduce the computational cost. In this approach, the environment is represented by a small number of key frames. A further improvement for [135] is in [144] with edge features. Endres *et al.* [143] provide an evaluation SLAM system using RGB-D camera. Mur-Artal *et al.* [136] design a novel system that uses the same features from all SLAM tasks, and propose a strategy to select the fittest points and keyframes for the reconstruction, to improve the robustness and generate a compact and trackable map. [136] further extends from the monocular camera to stereo and RGB-D cameras in [145].

For the latter category, it attempts to retrieve a more complete description of the environment at the cost of approximations compared to the inference methods. Some works rely on alternating optimization of pose and map by discarding cross-correlation of the estimated quantities [139] and [140]. Bloesch *et al.* [138] present an efficient representation of the scene by using a learned encoding to represent the dense geometry of a scene with small codes which can be efficiently stored and jointly optimized in multi-view SLAM.

2.2.0.4 Sensor fusion for SLAM

As each individual type of sensor has its limitations, for example LiDAR cannot see through small particles (i.e. smoke, fog or dust), camera is sensitive to the illumination variation, and UWB cannot measure the range correctly under NLOS, sensor fusion for SLAM is a proper way because each type of sensor has its strong points and limitations, thus they can complement the drawback of others. Due to this reason, some fusion based SLAM systems have received extensive attentions and been proposed recently among the literature. For example, Paul *et al.* [146] propose to fuse LiDAR data with Radar data to handle the SLAM problem in the environments with low visibility. The fusion takes place on scan level and map level to maximize the map quality considering the visibility situation. Janis *et al.* [127] present a fusion of monocular SLAM and UWB to widen the coverage of existing UWB localization system, and a smooth switch between UWB SLAM and monocular SLAM is constructed. Will *et al.* [147] propose a probabilistic model to fuse sparse 3D LiDAR data with stereo images to provide accurate dense depth maps and uncertainty estimates in real-time. Wang *et al.* [58] proposes to fuse UWB and visual inertial odometry to remove the visual drift with the aid of UWB ranging measurements thus improving the robustness of system. This system is infrastructure-based where our system is infrastructure-less. Perez *et al.* [148] propose a batch-mode solution to fuse ranging measurements from UWB sensor and 3D point-clouds from RGB-D sensor for mapping and localizing unmanned aerial vehicles (UAVs).

In this thesis, we also attempt to address the issue of sensor fusion (i.e. UWB and LiDAR) for SLAM under the same assumption in [148] that the locations of UWB beacons are priori unknown. Meanwhile, our system can simultaneously localize the robot, UWB beacons and build map in real-time whereas [148] proposes to collect all the data before start doing mapping and localization.

Our proposed fusion scheme for SLAM is different from the mostly related methods (i.e. [58, 148]) in the following aspects: a) The [148] can only estimate the UWB beacons' positions through offline processing, while the proposed SLAM runs in real-time. b) The scheme in [58] is infrastructure-based which demands some preparations such as anchors installation and calibration before SLAM starts. Notice that with our scheme the UWB beacons only need to be placed to the to-be-explored region but does not need to know beacons' positions so that our preparation effort is much less than that of [58].

2.3 Robotic Exploration System

The objective of robotic exploration is to navigate the robot through an unknown environment to execute a certain task (e.g. searching for victims), and construct a coarse/fine map for this environment. The former one highly depends on the on-board localization system, while the latter one requires a mapping module. To perform exploration, the robotic system needs to address the following issues: simultaneous localization and mapping (SLAM), determination of the intermediate points that define the robot's exploration behaviour (i.e. exploration scheme), and how to proceed from the current point to the next one (i.e. collision-free navigation). Depending on whether the robot is directed by a human or not, the exploration can be categorized into manual exploration and autonomous exploration. The former is usually regarded as a SLAM problem while the latter, a.k.a. autonomous exploration, needs to solve all problems as mentioned above. The literature works related to SLAM have already been elaborated in Section 2.2. In the following, we will briefly introduce the work involving robotic exploration.

2.3.1 Single-Sensor based Autonomous Exploration

Frontier-based approaches [5, 66, 67] are used to automate the exploration. Frontiers are the regions on the boundary between open space and unexplored space. By moving to new frontiers, a mobile robot can extend its map into new territories until the entire environment has been explored. In details, Yamauchi *et al.* [66] initially propose to detect the frontiers based on the evidence grid, while the exploration is occurred during the process of navigating the robot to these points. Later, occupancy grid map is used to represent the map and detect the frontiers in [149]. Holz *et al.* [67] evaluates the exploration efficiency of frontier-based approaches and reveal that the computational cost of detecting frontier edges increases rapidly as the explored map expanded. To handle such concern, some research works attempt to speed up the detection of frontier edges. Such as, Senarathne *et al.* [68] detect the frontiers by tracking the intermediate changes on grid cells, and the detection is only taken on the updated cells, thus reduce the searching space. Umari *et al.* [5] present an efficient frontier detection method based on a multiple rapidly-exploring randomized trees. Further, Niroui *et al.* [150] analyze the potential noisy of the sensory data, which may result in an inaccurate detection on frontiers, especially in the cluttered region. A partially observable Markov Decision Process method is proposed to deal with these uncertainty from sensors.

All the above approaches rely on either LiDAR or Camera. However, single-sensor based exploration system are prone to accumulating errors when constructing the map. To address this issue, the robot is allowed to revisit the explored regions to generate loop closures as mentioned in Section 2.2. For instance, [151] proposes a criterion combining the match likelihood and cost of revisiting previous locations to allow the robot to generate loop closures when necessary. However, the process of generating loop closures obviously sacrifices the exploration efficiency. Instead of generating loop closures, we attempt to eliminate accumulated errors by fusing two sensors, i.e., LiDAR and UWB. With the information estimated from UWB, the exploration scheme is not moving to frontiers, instead the robot in our system selects a UWB beacon to move to until it arrives in the beacon's vicinity and the

selection depends on how well the potential route is explored (less explored route is favoured).

2.3.2 Autonomous Exploration with Background Information

Background information such as a layout or a terrain map can facilitate the exploration task. This information can be estimated on the fly. For example, Sofman *et al.* [152] propose to create a top-view of a priori terrain map from the view point of an unmanned helicopter, and then utilize it to plan a feasible path for the ground vehicle; Ström *et al.* [153] infer the background information from previously experienced scenarios to facilitate the robotic system to make a better exploration behaviour, meanwhile a robust homing function is built to navigate the robot back to its starting location when the mapping system fails due to accumulated error. Schulz *et al.* [70] utilize a specifically trained text recognition system to read the door labels and sketch an abstract map for the environment with these found symbolic spatial information, which is then applied to guide the robot to the given destination. Further, a similar work is done in [154], but endows the robot a bit of reason abilities to understand the unknown environment. In [60], an active interaction is built between an unmanned aerial vehicle (UAV) and an unmanned ground vehicle (UGV), so that the UAV which has better view guides the UGV to do exploration. On the other hand, some exploration schemes require a priori background information before carrying out the exploration task. For example, Oßwald *et al.* [69] use the rough structure of the unknown space to optimize the intermediate points selection, and thus speed up the exploration process; Stachniss *et al.* [155] apply machine learning techniques to classify the environment into the semantic expression (i.e. a corridor or a room), and use these semantic information to help the robot avoid redundant works.

In this thesis, the region-of-interest is enclosed by UWB beacons whose locations are estimated while exploration is on-going. Meanwhile, the background information about the unknown environment can be reasoned from the UWB network and thus applied to assist the exploration.

Chapter 3

Event-Triggered Object Tracking for Long-term Visual Tracking

3.1 Introduction

In this chapter, we study the long-term object tracking topic and present an event-triggered tracking approach which attempt to detect the model drift and recover the tracker when such drift occurs. Motivated by the discussions in Section 1.1.1.1 and also inspired by the idea of event-triggered control proposed in control engineering [46, 47], an event-triggered tracking (ETT) framework with an effective occlusion and model drift identification approach is proposed in this chapter, which aim to simultaneously ensure fast and robust tracking with the required accuracy. A flowchart of the proposed tracker is given in Fig. 3.1. Note that the proposed tracking algorithm is not simply to employ short-term tracker interleaved with detection model. Also as seen from the detail comparisons presented in Table 3.1, the proposed approach has distinguishing features compared with existing drift-alleviating algorithms, which are reviewed and discussed in Section 2.

The proposed tracker is mainly composed of the following five modules: a short-term tracker, occlusion and drift identification, target re-detection, short-term

tracking model update and on-line discriminative learning for detector. An event-triggered decision model is built as the core component that coordinates these five modules. The short-term tracker is used to efficiently locate the target based on transformation information among the continuous frames. The occlusion and drift identification module evaluates current tracking state. If model drift is detected, the corresponding event will trigger the re-detection module to detect and reinitialize short-term tracking model. The tracker model updating is carried out at each frame, while the learning rate is dependent on the degree of occlusion. A sampling-pool is constructed to store discriminative samples and use them to update detector model.

The main features of the proposed approach and contributions of this chapter are summarised as follows.

1. A systematic ETT framework is built by decomposing the visual tracker into multiple event-triggered modules which work independently and are driven by particular events. It also allows future upgrade so as to introduce more sophisticated models.
2. An occlusion and drift detection method is proposed by considering the temporal and spatial tracking loss. It enables the event-triggered decision model to accurately evaluate the short-term tracking performance, and to trigger the relevant module as needed.
3. A bias alleviation sampling approach for SVM model is proposed. A sampling-pool with support samples, high confidence samples, and re-sampling samples, is constructed to alleviate the influence of noisy or mislabeled samples.
4. Weighted learning for short-term tracking model is applied, which will learn partial occlusions weakly and reject the heavy occlusion samples completely.

With such a novel ETT framework, the tracker is able to obtain a more robust short-term tracking performance and also properly address model drift problem

FIGURE 3.1: A framework of the proposed tracking algorithm. The correlation tracker is used to track the target efficiently. The tracker records the past predictions to discriminate the occlusion and model drift. The event-triggering model is used to produce triggering signals, labeled in different colors, to activate the corresponding subtasks, respectively. The detection model is used to re-detect the target when model drift happens.

for long-term visual tracking. Our proposed scheme is tested on frequently used benchmark datasets OTB-100 [2]. From the experimental results presented in Section 3.4, the proposed tracking scheme yields significant improvements over the state-of-the-art trackers under various evaluations conditions. More importantly, the proposed tracker runs at real-time and is much faster than most of the online trackers such as those in [36, 39, 43, 44, 48, 49].

3.1.1 Technical Comparisons Among Similar Trackers

Although the idea of fusion tracking, learning, and detection together is not new in the literature, how to incorporate them properly to improve the tracking performance is still a challenging problem. Different from similar state-of-art trackers, the proposed approach uses the concept of event-triggering to build a novel tracking framework, in which each module works independently and is only driven by its corresponding event. To achieve this, an occlusion and model drift detection model is constructed by measuring spatial and temporal tracking loss. Detailed technical

comparisons are listed in Table 3.1. According to the experimental results presented in Section 3.4, the proposed tracker achieves significant improvement over the listed trackers in terms of both tracking accuracy and speed. Especially, even though LCT tracker discussed in the introduction has similar modules to the proposed tracker, as listed in Table 3.1, the tracking accuracy of the proposed ETT is significantly higher than that of LCT while the tracking speed is almost the same.

TABLE 3.1: Technical comparisons among similar state-of-art trackers.

	TLD[42]	LCT[44]	MUSTer[43]	ROT[45]	MDP[115]	ETT
occlusion identification	×	×	×	✓	×	✓
drift identification	×	✓	×	×	✓	✓
object detection	✓	✓	✓	×	✓	✓
re-detection on demanded	×	✓	×	×	×	✓
weighted learning with occlusion	×	×	×	×	×	✓

3.2 Existing Techniques Used in Our Proposed Tracker

In this section, we present some preliminaries about correlation filter tracking and online-SVM classifier.

3.2.1 Techniques in Correlation Filter-based Tracking

Recently, correlation filter-based trackers [34–36, 38, 43, 44] have enhanced the robustness and efficiency of visual tracking. Generally, a translation correlation

filter F_t and scale correlation filter F_s are integrated together to perform translation and scale estimation respectively.

To perform translation estimation, a typical correlation filter F_t is trained on image patch $\mathbf{x}$ of $M \times N$ pixels, where all the cyclic shifts of $\mathbf{x}_{m,n}$ are training samples with $\mathbf{y}_{m,n}$ being the regression target for each $\mathbf{x}_{m,n}$, $(m, n) \in \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\}$. The goal is to find a function $g(\mathbf{z}) = \mathbf{w}^T \mathbf{z}$, where $\mathbf{w}$ is determined in such a way that the following objective is achieved:

$$\min_{\mathbf{w}} \sum_{m,n} |\langle \mathbf{w}, \phi(\mathbf{x}_{m,n}) \rangle - y_{m,n}|^2 + \lambda \|\mathbf{w}\|^2 \quad (3.1)$$

where ϕ denotes the mapping to a kernel space and $\lambda \geq 0$ is a regularization parameter.

Expressing the solution $\mathbf{w}$ as $\sum_{m,n} \alpha_{m,n} \phi(\mathbf{x}_{m,n})$ converts the optimization problem from variable $\mathbf{w}$ to $\boldsymbol{\alpha}$. The coefficient $\boldsymbol{\alpha}$ can then be obtained based on the properties of circulant matrices:

$$\boldsymbol{\alpha} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle) + \lambda} \right) \quad (3.2)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Fourier transform and its inverse, respectively. From the properties of Fourier transform, we can compute the filtering response for all candidate patches by substituting (3.2) to $g(\mathbf{z})$

$$\hat{f}(\mathbf{z}) = \mathcal{F}^{-1}(\mathcal{F}(\langle \phi(\mathbf{z}), \phi(\mathbf{x}) \rangle) \odot \mathcal{F}(\boldsymbol{\alpha})) \quad (3.3)$$

where $\odot$ denotes the element-wise product and $\hat{g}(\mathbf{z})$ is an estimation of $g(\mathbf{z})$. The translation is estimated by finding the patch with the maximal value of $\hat{g}(\mathbf{z})$ through (3.3).

To handle the scale problem, a one-dimensional correlation filter F_s is trained on Q image patches, where pyramid around the estimated translation location is cropped from the image. Assume that the target size in the current frame is $W \times H$ and

let P denote the account of scales $s \in \{\varepsilon^p | p = \lfloor -\frac{P-1}{2} \rfloor, \lfloor -\frac{P-3}{2} \rfloor, \dots, \lfloor \frac{P-1}{2} \rfloor\}$ where ε is a parameter representing the base value of scale changing. Then all Q image patches I_s with size $sW \times sH$ centered around the estimation location are resized to the template size for feature extraction. The optimal scale $\hat{s}$ is given as the highest correlation filter response of image patch I_s .

3.2.2 Techniques in Discriminative Online-SVM Classifier

We consider a linear classifier of the form $f(\mathbf{x}|\boldsymbol{\omega}, b) = \boldsymbol{\omega}^T \mathbf{x} + b$, where $\boldsymbol{\omega}$ is the weight vector, b is the bias threshold. Suppose we have one previously trained model and a set of new training samples $\mathbf{x}_i \in \mathbb{R}^n$. The online learning algorithm trains the model at trial t is confined to use samples from past t trials, which can be regarded as solving the following optimization problem

$$\min_{\boldsymbol{\omega}} \frac{1}{2} \|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^{t-1} \ell^\gamma(\boldsymbol{\omega}; (\mathbf{x}_i, y_i)) \quad (3.4)$$

where $y_i \in \{1, -1\}$ is the binary label of $\mathbf{x}_i$, parameter C controls the trade-off between the complexity of $\boldsymbol{\omega}$ and the cumulative *hinge-loss* of $\boldsymbol{\omega}$, ℓ^γ is the *hinge-loss* function [156].

Using the notion of duality, (3.4) can be converted into its equivalent dual form as

$$\begin{aligned} \max_{\boldsymbol{\beta}} \mathcal{D}_t(\boldsymbol{\beta}) &= \sum_{i=1}^{t-1} \beta_i - \frac{1}{2} \sum_{i=1}^{t-1} \sum_{j=1}^{t-1} \beta_i \beta_j y_i y_j K_{ij} \\ 0 \leq \alpha_i \leq C, \quad &\sum_{i=1}^{t-1} \beta_i y_i = 0, \end{aligned} \quad (3.5)$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is a symmetric positive definite function, such as Gaussian function.

Therefore, the online-SVM algorithm can be treated as an incremental solver of dual problem [156], where at the end of trial t the algorithm maximizes the dual

function confined to the first $t - 1$ observed variables. Finally, the optimal parameter ω could be easily obtained by using existing convex optimization tools, such as LIBSVM [157].

Algorithm 1: The proposed autonomous exploration scheme

Require: initial bounding box r_0 , image sequence $\{I_t\}_0^T$

```

1: Initialize correlation filter and online-SVM classifier by using  $B_0^e$  and  $I_0$ 
2: for  $t = 1 : T$  do
3:    $B_c \leftarrow$  correlation filter prediction bounding box
4:    $\mathcal{L}_{x^\circ, p^\circ}^t \leftarrow \delta \mathcal{L}_{x^s}^S + (1 - \delta) \mathcal{L}_{p^\circ}^T$  {Triggering each events}
5:   if  $\mathcal{L}_{x^\circ, p^\circ}^t > T_{fail}$  then
6:     Do object re-detection by finding  $B_r \leftarrow \max_{h_i} \omega^T \Phi(B_i) + b$ 
7:     Reinitialize correlation filter by patch  $(I_t, B_r)$ 
8:      $B_t^e \leftarrow B_r$  & Continue tracking next frame
9:   end if
10:  if  $T_{occ} < \mathcal{L}_{x^\circ, p^\circ}^t < T_{fail}$  then
11:     $\xi_o \leftarrow \xi_n \frac{T_{fail} - \mathcal{L}_{x^\circ, p^\circ}^t}{T_{fail} - T_{occ}}$ 
12:  end if
13:  if  $\mathcal{L}_{x^\circ, p^\circ}^t < T_{occ}$  then
14:     $\xi_o \leftarrow \xi_n$ 
15:    Do re-sampling for update detector model
16:    if  $\mathcal{L}_{x^\circ, p^\circ}^t < T_{ap}$  and  $m_{rs} > \epsilon$  then
17:      Do detecting model update
18:    end if
19:  end if
20:  Do correlation filter update
21:   $B_t^e \leftarrow B_c$ 
22: end for
23: return estimated bounding boxes  $\{B_t^e\}_1^T$ 

```

3.3 The Proposed Techniques

In this section, the details of newly proposed techniques including occlusion and model drift detection and event-triggering conditions are given with reference to Fig. 3.1.

3.3.1 Occlusion and Model Drift Detection

One of major issues in visual tracking is to handle model drift and also occlusion, as they are prone to result in tracking failure. Factors leading to model drift include illumination variation, fast movement, appearance change and occlusion. It should be noted that occlusion can be considered as a separate issue for its generality and challenge in visual tracking. The drift problem could also be alleviated by adjusting the learning strategy of short-term tracking model when occlusion occurs.

To better assess the short-term tracking process, two criteria are defined and then fused to verify current tracking state:

Criterion 1 (Spatial locality-constraint). For any accurate tracking frame, a patch which is closer to the center of the prediction should be more similar to the target, and vice versa. Otherwise, this criterion cannot be met if the present tracking is incorrect or in the presence of occlusion or drift.

Denote $\mathbb{E}_S(\cdot, \cdot)$ as a metric that defines the similarity between two patches, c_{gt} is the center of ground truth patch, c_i and c_j are the centers of candidate patches. According to Criterion 1, an accurate tracking frame should satisfy the following condition.

$$\mathbb{E}_S(c_{gt}, c_i) < \mathbb{E}_S(c_{gt}, c_j) \text{ if } \|c_{gt} - c_i\|_2 < \|c_{gt} - c_j\|_2 \quad (3.6)$$

Criterion 2 (Temporal continuity). The cumulative estimation loss for short-term tracking within a recent temporal window should be close to zero if the predictions given by short-term tracker are accurate or without drift, and vice versa.

Denote $\mathbb{E}_T(\cdot, \cdot)$ as a metric for measuring estimation loss. $P(\cdot)$ and $P'(\cdot)$ are two prediction models, $P_{gt}(\cdot)$ denotes the ground truth. From Criterion 2, an accurate tracking in a predefined time window should obey the following condition if $P(\cdot)$ is more accurate than $P'(\cdot)$.

$$\sum_{i=0}^{\Delta} \mathbb{E}_T(P(t-i), P_{gt}(t-i)) < \sum_{i=0}^{\Delta} \mathbb{E}_T(P'(t-i), P_{gt}(t-i)) \quad (3.7)$$

where t is the number of tracking frame. Note that the cumulation loss is computed in a time window to achieve a robust loss estimation.

Based on Criteria 1 and 2, two tracking losses are proposed and fused to evaluate the short-term tracking performance.

3.3.1.1 Spatial Loss Evaluation

Following Criterion 1, a spatial loss evaluation is designed. Denote all the possible candidate patches within a square neighborhood of the t -th prediction patch $x_{p,t}^\circ$ as $\mathbf{x}^\circ = \{I_t, u_i, v_i, w_{p,t}, h_{p,t}\}_{i=1}^{N_s}$, where I_t is the image of the t -th frame, (u_i, v_i) denotes the position, $(w_{p,t}, h_{p,t})$ represents the scale of current prediction $x_{p,t}^\circ$, N_s is the number of patches. A real-valued linear function $f(\Phi(x_i^\circ)|\boldsymbol{\omega}, b) = \boldsymbol{\omega}^T \Phi(x_i^\circ) + b$ with details given in Section 3.2.2 is employed to estimate the confidence score for each patch x_i° , where Φ is a mapping from the original input space to the feature space. Meanwhile, the label for x_i° is produced by calculating the overlap rate with $x_{p,t}^\circ$.

$$L^\circ(x_i^\circ|x_{p,t}^\circ) = \frac{\text{area}(ROI_{x_i^\circ} \cap ROI_{x_{p,t}^\circ})}{\text{area}(ROI_{x_i^\circ} \cup ROI_{x_{p,t}^\circ})} \quad (3.8)$$

In this chapter, we use Kullback-Leibler divergence [158] to assess spatial loss defined as:

$$\mathcal{L}_{\mathbf{x}^\circ}^S = \sum_{x_i^\circ \in \mathbf{x}^\circ} \bar{f}(\Phi(x_i^\circ)|\boldsymbol{\omega}, b) \log \frac{\bar{f}(\Phi(x_i^\circ)|\boldsymbol{\omega}, b)}{\bar{L}^\circ(x_i^\circ)} \quad (3.9)$$

where $\bar{f}$ and $\bar{L}^\circ$ are the normalized form.

3.3.1.2 Temporal Loss Evaluation

Such an evaluation is designed according to Criterion 2. The main objective of the temporal loss function is to evaluate the accuracy of the short-term tracker in a recent time period. Let the desired confidence score be L^T . Denote the estimated patches of recent Δ predictions as $\mathbf{p}^\circ = \{I_i, u_i, v_i, w_i, h_i\}_{i=t-\Delta}^t$, where Δ is the size of the temporal window. The temporal loss function is defined as:

$$\mathcal{L}_{\mathbf{p}^\circ}^T = \frac{1}{\Delta + 1} \sum_{i=t-\Delta}^t \varpi_{t,i} \|f(\Phi(p_i^\circ)|\boldsymbol{\omega}, b) - L^T\|_2^2 \quad (3.10)$$

where $\|\cdot\|_2$ is 2-norm, $\varpi_{t,i}$ denotes the weight for each past prediction, such as forgetting curve $e^{-\frac{t}{\rho}}$ and uniform distribution. Set L^T for each prediction to +1 since the confidence $f(\Phi(x_i^\circ)|\boldsymbol{\omega}, b)$ should be close to +1 if current prediction is accurate.

Finally, the tracking loss at the t -th frame is the fusion of spatial loss and temporal loss.

$$\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t = \delta \mathcal{L}_{\mathbf{x}^s}^S + (1 - \delta) \mathcal{L}_{\mathbf{p}^\circ}^T \quad (3.11)$$

where $\delta \in [0, 1]$, which is a weighting factor to balance the spatial loss and temporal loss.

A smaller tracking loss represents better tracking performance. Two thresholds T_{fail} and T_{occ} are initialized to classify the tracking state into normal tracking, partial occlusion, and drift/heavy occlusion. The tracking state is set to drift if $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{fail}$, or partial occlusion if $T_{fail} \geq \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{occ}$, otherwise set to normal tracking. However, a fixed threshold may not be an appropriate way for various video sequences. In this chapter, a threshold learning approach is proposed to update T_{fail} and T_{occ} adaptively by considering the recent tracking losses $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^i\}_{i=t-\Delta}^t$ in relation to the current threshold. Let θ_d and θ_o denote the learning rate for T_{fail} and T_{occ} respectively. The adaptive thresholds are defined as

$$T = T + \theta \left(\frac{1}{\Delta + 1} \sum \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ} - T \right) \quad (3.12)$$

FIGURE 3.2: Illustration for drift detection and restoration. The bounding boxes colored in green, red solid, red dash and yellow denote ground truth, ETT prediction, the prediction without re-detection and corrected results after detecting tracking failure, respectively.

where $T = \{T_{fail}, T_{occ}\}$ and $\theta = \{\theta_d, \theta_o\}$. An upper and a lower bound are set for T to avoid adaptive threshold becoming too large or too small. Fig. 3.2 illustrates the process of drift detection and restoration. The black line denotes the tracking loss for each frame, and the blue line is the tracking drift threshold, which is adaptively changed according to (3.12).

3.3.2 Event-triggered Decision Model

As previously mentioned, the tracker consists of multiple modules, each of which is intended to conduct a subtask of the entire visual tracking. An event for each module is associated with a set of conditions. When those certain conditions for an event are met, the event becomes active and triggers the corresponding subtask to be executed. The details of the proposed events are summarized in Table 3.2.

TABLE 3.2: The feature representation for the defined events.

Event	Notation	Function Description
Correlation Tracking Model Updating 1 (CTMU1)	$\mathbf{E}_{ctmu1}$	Updating the correlation tracking model when no model drift or occlusion happens
Correlation Tracking Model Updating 2 (CTMU2)	$\mathbf{E}_{ctmu2}$	Updating the correlation tracking model when partial occlusion happens
Heuristic Object Re-detection (HOR)	$\mathbf{E}_{hor}$	Re-detecting the target when model drift happens
Re-sampling for Detector Model (RDM)	$\mathbf{E}_{rdm}$	Extracting the new positive and negative samples based on current frame and pushing them to the sampling-pool when the prediction is accurate
Detector Model Updating (DMU)	$\mathbf{E}_{dmu}$	Updating the detector model when the model is not suitable for current appearance
Normal Tracking (NT)	$\mathbf{E}_{nt}$	Continuing the short-term tracking when no drift is detected

Let $\mathbf{E}$ denote the event set, i.e $\mathbf{E} = \{\mathbf{E}_{ctmu1}, \mathbf{E}_{ctmu2}, \mathbf{E}_{rdm}, \mathbf{E}_{hor}, \mathbf{E}_{dmu}, \mathbf{E}_{nt}\}$. Each event in the set is independent of the others and has two states, i.e active or inactive, represented by 1 or 0, respectively, to indicate the executing status of its corresponding subtask. If all the conditions of an event are met, then its state turns to active. For instance, the state of $\mathbf{E}_{hor}$ being active indicates that model drift is detected, then the re-detection module will be triggered to relocate the target, otherwise re-detection module is in inactive (sleeping) state. During the tracking process, the aforementioned six events are verified sequentially and the state for each event is set accordingly.

It is essential to cooperatively integrate these events such that the tracker is able to carry out the entire task efficiently and robustly. In this chapter, we use the information collected from occlusion and drift diagnosis model, as detailed in Section 3.3.1, and a decision tree to build an event-triggered decision model, as shown in Fig. 3.3.

FIGURE 3.3: Illustration of the event-triggered decision tree. The event-triggered decision model will produce one or multiple events based on the input from occlusion and drift identification model, which provides the information of short-term tracking state evaluation.

3.3.2.1 Correlation Tracking Model Updating

The correlation tracking state could be classified into normal tracking, tracking with partial occlusion and tracking with model drift or heavy occlusion. Generally, model updating for tracker and detector should be carried out when the first two situations are met but not for model drift or heavy occlusion, since noisy samples will cause model pollution. However, long-term partial occlusion also has a great impact on the model, thus using a learning rate smaller than that in normal tracking is necessary. According to the discussions in Section 3.3.1, the corresponding events for updating short-term tracker on the situation of normal tracking and partial

occlusion are proposed respectively as

$$\begin{cases} \mathbf{E}_{ctmu1} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{occ}\} \\ \mathbf{E}_{ctmu2} = \{T_{occ} \leq \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{fail}\}. \end{cases} \quad (3.13)$$

Based on the aforementioned observations, the learning rate for correlation tracker is computed as:

$$\xi_o = \begin{cases} \xi_n, & \text{if } \mathbf{E}_{ctmu1} = 1 \\ \xi_n \frac{T_{fail} - \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t}{T_{fail} - T_{occ}}, & \text{if } \mathbf{E}_{ctmu2} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.14)$$

where ξ_n is the learning rate for normal tracking. The learning rate on the situation of partial occlusion decreases linearly from ξ_n according to occlusion degree.

Therefore, the translation filter F_t and scale filter F_s in Section 3.2.1 could be updated with the learning rate ξ_o as

$$\begin{cases} \hat{\mathbf{x}}^t = (1 - \xi_o)\hat{\mathbf{x}}^{t-1} + \xi_o\mathbf{x}^t \\ \hat{\boldsymbol{\alpha}}^t = (1 - \xi_o)\hat{\boldsymbol{\alpha}}^{t-1} + \xi_o\boldsymbol{\alpha}^t \end{cases} \quad (3.15)$$

where $\hat{\mathbf{x}}^t$ and $\hat{\boldsymbol{\alpha}}^t$ are the learned object appearance and parameter of correlation filter, respectively. They are considered as the respective estimation of $\mathbf{x}^t$ and $\boldsymbol{\alpha}^t$, and thus used to determine $\hat{g}(\mathbf{z})$ in (3.3).

3.3.2.2 Heuristic Object Re-detection

The object re-detection module is triggered when tracking failure is detected. Thus the event $\mathbf{E}_{hor}$ is proposed as follows:

$$\mathbf{E}_{hor} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{fail}\} \quad (3.16)$$

In this case, most existing short-term trackers cannot recover from drift since the model is contaminated due to inappropriate updating on noisy samples. To overcome this problem, a new discriminative appearance model should be used to relocate the target and reinitialize the currently polluted short-term tracker.

In the proposed tracking algorithm, we implement a discriminative online-SVM classifier detailed in Section 3.2.2 as our detector. Given a frame I , the state of a sample is represented by a bounding box $\mathbf{B} = \{I_t, u_i, v_i, w_i, h_i\}_{i=1}^{N_b}$. The feature extracting from sample B_i is denoted as $\mathbf{h}_i = \Phi(B_i)$. Then the detection task is converted to the following problem:

$$\max_{\mathbf{h}_i} f(\mathbf{h}_i | \boldsymbol{\omega}, b) = \max_{\mathbf{h}_i} \boldsymbol{\omega}^T \Phi(B_i) + b \quad (3.17)$$

where $\boldsymbol{\omega}$ and b are the optimized parameters of SVM model.

As the number of bounding boxes N_b to be evaluated is large for the holistic frame, the classification process requires a huge amount of computation. However, the detection process could be accelerated based on the past tracking performance, which provides important information about the area where the target possibly locates. Let $\mathcal{C}(b_c, r)$ denote the circular region with center point b_c and radius r . Suppose that r_b is a predefined basic radius. Then according to mean value of past tracking loss, r could be adaptively adjusted as

$$r = r_b \left(1 + \frac{1}{\Delta + 1} \sum_{i=t-\Delta}^t \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^i \right) \quad (3.18)$$

because the bigger the mean tracking loss indicates, the higher the possibility that the target is lost or far away from current prediction. By considering the *priori* tracking information, the detector could always find an optimal searching region without sacrificing accuracy.

3.3.2.3 Detector Model Updating

A robust tracking model normally needs to be updated continuously based on current prediction to capture the appearance variation of the target. However, this semi-supervised learning has its drawbacks. Firstly, it is sensitive to new samples. The noisy or mislabeled samples will cause catastrophic results to the original model. Secondly, due to continuous accumulation of new samples, the sampling-pool would become increasingly larger, which significantly slows down the on-line learning process.

To overcome the above problems, the following sampling-pool $\mathcal{T}$ is constructed to robustly update SVM model.

$$\mathcal{T} = \mathcal{T}_{sup} \cup \mathcal{T}_{conf} \cup \mathcal{T}_{rs} \quad (3.19)$$

where $\mathcal{T}_{sup}$ denotes support samples from previous SVM model, $\mathcal{T}_{conf}$ is the set of samples with high confidence of being positive or negative, which is obtained by collecting the n_{sup} farthest samples from the model margin. The primary objective of $\mathcal{T}_{conf}$ is to keep the model robust to noisy or mislabeled samples. $\mathcal{T}_{rs}$ is the set of new samples sampled between previous training stamp and current training stamp. The maximum capacities of $\mathcal{T}_{conf}$ and $\mathcal{T}_{rs}$ are set as M_{conf} and M_{rs} respectively. When the maximum capacity is reached, some samples need to be removed randomly. Set $\mathcal{T}_{rs}$ will be reset to empty once the SVM model is retrained. Fig. 3.4 illustrates the principle of sampling-pool.

It is unnecessary to update the detector model as frequently as short-term tracking model since the detector model is not sensitive to small variations of appearance among consecutive frames, while the retraining process is time-consuming. Generally, the updating should be executed when the model is not suitable for the current appearance, which may be caused by, for example, large appearance variation. The reasons are: 1) samples with small appearance change have a high probability of locating outside the margin or overlapping with existing support vectors based

on the Karush-Kuhn-Tucker (KKT) conditions [159], which have little impact on SVM model. 2) the samples with high appearance variation are likely to reside around the margin of SVM model, which would have a remarkable influence on the current model and make it more generalized. With the above observations, and

(a) The composition of positive and negative sampling-pool

(b) Updating for sampling-pool

FIGURE 3.4: Illustration of the representation: (a) shows the composition of sampling-pool, which is divided into three portions, namely, support samples, high confidence samples and re-sampled samples; and (b) shows the updating methods for the first two parts in (a).

letting $m_{rs} = m_{rs+} + m_{rs-}$, which is the current size of $\mathcal{T}_{rs}$, an event $\mathbf{E}_{dmu}$ is now proposed to activate the updating of SVM model as follows:

$$\mathbf{E}_{dmu} = \{\mathcal{L}_{x^\circ, p^\circ}^t < T_{ap} \cap m_{rs} > \epsilon\} \quad (3.20)$$

where ϵ and T_{ap} are predefined constants. T_{ap} is a threshold for huge appearance changing.

3.3.2.4 Re-sampling for Detector Model

Re-sampling happens when triggered by the event defined in (3.21). Then the event $\mathbf{E}_{rdm}$ is proposed as

$$\mathbf{E}_{rdm} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{occ}\} \quad (3.21)$$

which means that the current prediction is accurate. In this case, the tracker will sample n_{rs+} patches around prediction as positive samples and n_{rs-} patches far away from prediction as negative samples, which are then added to $\mathcal{T}_{rs}$. When $\mathcal{T}_{rs} \geq M_{rs}$ is full, some redundant samples will be removed from the sampling-pool $\mathcal{T}_{rs}$.

Normal Tracking The event representing normal tracking is defined as

$$\mathbf{E}_{nt} = \mathbf{E}_{ctmu1} \cap \mathbf{E}_{ctmu2} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{fail}\} \quad (3.22)$$

It also indicates that there is no model drift detected. In this case, the short-term tracker could continuously perform tracking task even with partial occlusion, and the re-detection module will not be activated for the disturbance from environment is under the tolerance of short-term tracker. In this case, the prediction given by short-term tracker is still considered reliable. Overall, the proposed method is illustrated in Algorithm 1.

3.4 Experiments

3.4.1 Implementation Details

Two different types of features are applied to the proposed tracking algorithm. Specifically, HOG features with 31 bins and 4x4 cell size are used in the short-term tracker, while 300 Haar-like features with 6 different types and 50 bounding boxes are adopted to train and test the online-SVM classifier. The parameters for the proposed ETT tracker are specified as follows: The parameters for correlation filter based short-term tracking is the same as in [38]. The number of past predictions Δ is used to detect occlusion and tracking failure/drift, which should neither be too small to make drift detection sensitive, nor too big to make drift detection unresponsive or dullness. Empirically, $\Delta = 20$ will provide a quite satisfactory performance. The initialized occlusion threshold T_{occ} and failure/drift threshold T_{fail} are set as 0.3 and 0.6 respectively. The lower bounds for T_{occ} and T_{fail} are set as 0.2 and 0.4, respectively. and upper bounds are 0.4 and 0.8, respectively. The learning rate θ_d and θ_o for T_{fail} and T_{occ} are set to 0.01 and 0.015 respectively. The threshold for huge appearance changing T_{ap} is set to 0.25. The value of δ is set to 0.45. Empirically, there is little influence for tracking accuracy when δ is set between 0.3 to 0.7. In all the experimental tests with different schemes, these parameters are set to be the same for the whole testing sequences.

(a) Results for OTB-50 [1] dataset

(b) Results for OTB-50(hard) sequences in [2]

(c) Results for OTB-100 sequences in [2]

FIGURE 3.5: Quantitative results on the benchmark datasets. The scores in the legends indicate the average Area-Under-Curve value for precision and success plots, respectively.

TABLE 3.3: Comparisons among baseline trackers.

	OTB-50(hard) [2]		OTB-100 [2]		Mean FPS
	DP (%)	OP (%)	DP (%)	OP (%)	
DSST [38]	57.7	46.4	63.0	52.0	37.7
ETT_{RD}	59.4	49.1	66.0	55.2	27.0
ETT_{FD}	66.9	54.5	73.6	61.1	6.5
ETT_{DU}	67.8	55.4	73.7	61.4	21.0
ETT	71.3	59.0	76.2	63.7	18.1

3.4.2 Evaluation on OTB-50 [1] and OTB-100 [2] Dataset

TABLE 3.4: A time consumption comparison. The mean overlap precision (OP) (%), distance precision (DP) (%) and mean FPS over all the 100 videos in the OTB-100 [2] are presented. The two best results are displayed in red and blue, respectively.

	Mean OP(%)	Mean DP(%)	Mean FPS
ETT(ours)	63.7	76.2	18.1
MUSTer	58.9	72.1	5.0
LCT-Deep	59.4	76.3	17.9
SRDCF	60.0	71.8	6.6
LCT	57.4	69.9	17.1
MEEM	53.9	71.1	13.5
Staple	58.1	71.3	17.6
DSST	52.0	63.2	37.7
KCF	49.1	64.6	245
STRUCK	46.4	59.6	10.5
TLD	40.9	52.5	25.3
ROT	48.7	60.0	28.6

The OTB-50 [1] contains 50 videos. The OTB-100 [2] is expanded from OTB-50 [1] which contains 100 videos. In order to facilitate an in-depth analysis, the authors in [2] select 50 difficult and representative sequences from the whole dataset.

This subset is denoted as OTB-50(hard) [2]. Note that the 50 sequences in OTB-50(hard) are different from those in OTB-50[1].

3.4.2.1 Components Analysis

To evaluate the contribution made by each module, we test following variations of the proposed method with a combination of different modules. a) ETT_{BASE} : the baseline tracker of the proposed method. In our method, we implement DSST [38] as our baseline tracker. b) ETT_{RD} : the ETT tracker without target re-detection module. c) ETT_{FD} : the ETT tracker without occlusion and failure detection module which means detection will occur at each frame. d) ETT_{DU} : the ETT tracker without online discriminate learning for detector module, the detector model is trained by using samples collected from first 5 frames and then it is fixed in the following frames. e) ETT : the proposed overall event-triggered tracker.

As shown in Table 4.2, it can be concluded that 1) weighted learning for correlation filter tracker has provided certain ability to address occlusion when comparing ETT_{RD} with baseline tracker. 2) Occlusion and drift detection is useful when comparing ETT_{FD} with ETT . Due to the occlusion and drift detection module in ETT , the tracker could alleviate the influence from noisy samples and improve the accuracy. On the other hand, it greatly increases the tracking speed by re-detecting the target only when needed. 3) Comparing ETT_{DU} with baseline tracker, it can be concluded that re-detection is an effective way to restore tracking failure. ETT_{DU} has a weak detector since it only trains a detection model by the samples collected from first 5 frames, even though ETT_{DU} improves more than 9 percent compared with baseline tracker. 4) The online discriminate learning is helpful when comparing ETT with ETT_{DU} , since it could improve the adaptability of detector to handle various environment changing. With all the modules, the proposed tracker improves around 11% than the baseline tracker in terms of overlap precision.

3.4.2.2 Quantitative Evaluation

The proposed ETT approach is compared with eleven state-of-the-art trackers (MUSTer [43], LCT [44], LCT-Deep [116], SRDCF [49], Staple [36], MEEM [48], DSST [38], TLD [42], ROT [45], KCF [34] and STRUCK [39]) on OTB-50 dataset [1] and its extended version OTB-100 dataset [2]. The precision and success plots¹ over these datasets are presented in Fig. 3.5.

FIGURE 3.6: Quantitative results on OTB-100 [2] for 8 challenging attributes: motion blur, illumination variation, background clutters, occlusion, out-of-view, deformation, scale variation and out-of-plane rotation.

According to Fig. 3.5, it is observed that the proposed method achieves a similar accuracy on OTB-50 [1] compared with LCT-Deep [116] and MUSTer [43], while surpasses the state-of-the-art trackers both in accuracy and precision by a large margin on OTB-100 [2]. Especially on more challenging dataset OTB-50(hard) [2], the proposed method also achieves favorable performances when compared with other trackers. According to Fig. 3.5 (a) and (c), due to more challenging sequences added to the OTB-50 [1] dataset, including great deformation, long-term occlusion *et al.*, the performance of all other trackers are degraded while

¹The average Area-Under-Curve score is different with the score at a specific threshold, i.e. the distance precision score of MEEM in OTB-50 at the threshold of 20 pixels is 0.83 while the average AUC score is 0.743.

the proposed method almost maintains the same performance. This is owing to that the proposed failure detection and event-triggered target re-detection can promptly and effectively handle these challenging attributes. As shown in Fig. 3.5 (b) and (c), the proposed method outperforms the second best tracker (LCT-Deep and SRDCF) by almost 4 percent in terms of overlap success rate and distance precision.

3.4.2.3 Comparisons on different attributes on OTB-100

To thoroughly evaluate the robustness of the proposed ETT tracker in various scenes, we present tracking accuracy in terms of 8 challenging attributes on OTB-100 [2] in Fig. 4.8. As observed from the figure, the proposed tracker ETT outperforms other methods by a huge margin in all the attributes, particularly in handling occlusion, deformation, scale variation, out-of-plane rotation and out-of-view, which can be attributed to the effectiveness of event-triggered detection approach in relocating target when tracking drifts occur. It also demonstrates that the proposed event based triggering mechanism can accurately recognize tracking failure effectively. Note that the trackers MUSTer, LCT, LCT-Deep and TLD also include a detection module, however, they show their limitations in handling these challenging situations.

3.4.2.4 Comparisons on tracking speed

With the configuration in Section 4.3.1, we compare the speed of the 11 trackers on a platform with Intel Xeon E5-1630 3.70 GHz CPU and 16.0 GB RAM. The results for all these trackers are listed in Table 3.4, based on which the following observations are made. 1) It is seen that ETT tracker not only achieves the highest tracking precision but also operates in real-time. 2) Compared to the trackers LCT and MUSTer, which also adopt re-detection modules, ETT tracker improves significantly both on distance precision and overlap precision, and in the meantime, its tracking speed is faster than that of LCT and MUSTer. 3) Although trackers KCF

and DSST give faster tracking speed, their tracking accuracies are unsatisfactory. On the other hand, our ETT tracker improves significantly on tracking accuracy, while with a similar tracking speed when compared to MEEM, Staple, LCT, and SRDCF.

3.4.2.5 Evaluation on VOT16 [3] Dataset

In addition to OTB-100, we also evaluate the proposed method on VOT2016[3], which contains 60 challenging real-life videos. We compare our tracker with the following state-of-the-art methods: SCT4[160], SODLT [161], DNT[162], DSST2014[38], KCF2014[34], ANT[163], ASMS[164], TricTRACK[165], MAD[166], SiamAN[102], SRDCF[49], TGPR[167], Staple[36] and DeepSRDCF[49].

As shown in Fig. 3.7, the proposed tracker ranks first in terms of accuracy and third in terms of robustness. Among these competitive trackers, the deep learning based ones (SODLT[161], DNT[162], SiamAN[102] and DeepSRDCF[49]) generally achieve a higher rank in robustness. However, the proposed tracker achieves a good trade-off between robustness and accuracy which makes our methods competitive to the other trackers. Note that, the deep learning based trackers usually have a low tracking speed and require additional hardware support of GPU. Our proposed tracker runs on CPU only and can achieve a speed of 18 FPS, which makes it suitable for real-time applications.

These results demonstrate that our tracker is able to locate the targets in the complicated scenarios more precisely than the other top trackers. Overall, the proposed approach outperforms the state-of-the-art trackers regarding to overlap success and distance precision.

3.4.3 Discussion on Failure Cases

Generally, the performance of the proposed ETT strategy is determined by three modules: 1) occlusion and drift detection model, 2) target re-location model, 3)

FIGURE 3.7: Accuracy-robustness ranking plot for the state-of-the-art comparison.

short-term tracker. The proposed ETT may not perform well once one of these modules cannot work properly, such as for some extremely challenging cases like the *hand*, *fish1* in VOT16[3] and the *Diving*, *Jump* in OTB-100[2]. For the former sequences, the target is quite similar to interference of surroundings, resulting in that the ETT can hardly extract useful spatial and temporal loss information to verify current tracking performance. In such an environment, the proposed tracker could not effectively identify the drift cases since Criteria 1 and 2 are hardly satisfied. For the latter sequences, the ETT fails to restore the tracker from tracking drift due to the limitation of features used to describe the target and candidates. Therefore, we can conclude that the following cases will result in failure of ETT.

- the appearances of foreground and background are similar such that the occlusion and drift detection model cannot properly assess the performance of short-term tracker.
- the short-term tracker frequently fails to track the target, resulting in slow tracking speed of ETT.
- the target is very small or with less feature that the re-location model cannot properly locate the target when tracking failure occurs.

In fact, the aforementioned sequences are so challenging that most of the trackers cannot perform well. On the other hand, the comparison results on these two large datasets with various objects demonstrate that the proposed tracking algorithm is able to detect occlusion and drift accurately, which promptly triggers to address the tracking failure by re-detecting the target. As such, higher tracking accuracy is achieved while the tracking speed is also improved. In conclusion, based on the experimental results, the proposed ETT approach is proven to be sufficiently effective and efficient to handle various environmental challenges.

3.5 Conclusions

In this chapter, we present a simple but effective tracking framework by proposing event-triggered tracking algorithms with occlusion and drift detection. The overall tracker is decomposed into several independent modules involving different subtasks. An event-triggered decision model is proposed to coordinate those modules in various scenarios. Moreover, a novel occlusion and drift detection algorithm is proposed in this chapter to tackle the general yet challenging drift and occlusion problems. Our tracking algorithm outperforms the state-of-the-art trackers in terms of both tracking speed and accuracy, in OTB-100 [2] and VOT16 [3] benchmarks.

Chapter 4

Adaptive Multi-feature Reliability Re-determination Correlation Filter for Visual Tracking

4.1 Introduction

The tracking approach presented in Chapter 3 focuses on how to identify the model drift and thus recover the tracker when tracking failure is detected. The experimental results show that such a tracking framework works well for the cases of occlusion and out-of-view, but the target with deformation, background clutters or non-rigid rotation is still a challenge for it. One of the most important reasons is that the feature applied for tracking in Chapter 3 is too weak (i.e. HOG). Therefore, as discussed in Chapter 1, we attempt to alleviate the model drift by formulating various features (i.e. handcraft features and deep features) within one optimization framework which can make use of their respective advantages. In this chapter, we will present an adaptive reliability re-determination correlation filter, followed by two different solutions involving how to re-determine the reliability for each feature.

The contribution of this chapter and main features of the proposed method are summarized as follows:

1. We formulate a reliability re-determinative correlation filter which takes the importance of each feature into consideration when optimizes the CF model, thus enables the tracker to trust more on the features that are suitable for the current tracking scenario;
2. Two different solutions, named numerical optimization and model evaluation, are proposed to online re-determine the reliability for each feature. Meanwhile, two independent trackers are implemented based on the proposed two weight solvers.
3. Extensive experimental evaluations have been designed to validate the performance of the proposed two trackers on five large datasets, including OTB-100 [2], TempleColor [33], VOT2016 [3], VOT2018 [81] and LaSOT [168].

The experimental results demonstrate that the proposed reliability re-determination scheme can effectively alleviate the model drift. Especially on VOT2016, both trackers have achieved outstanding tracking results in terms of EAO score, which significantly outperform the recently published top trackers.

4.2 Reliability Re-determination Correlation Filter

In this section, we present the details of the proposed tracking framework.

Learning a discriminative correlation filter in spatial domain is formulated by minimizing the following objective function [34]:

$$E(\mathbf{h}) = \frac{1}{2} \|\mathbf{g} - \mathbf{X}\mathbf{h}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2, \quad (4.1)$$

where the vector $\mathbf{h} \in \mathbb{R}^N$ denotes the desired CF model. The square matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ is a combination of all circulant shifts of the vectorized image patch $\mathbf{x}$. N is the number of elements in the image patch $\mathbf{x}$. The vector $\mathbf{g} \in \mathbb{R}^N$ denotes the regression target, which is usually a vectorized image of a 2D Gaussian with small variance. λ is a regularization parameter.

The (4.1) is formulated for one type of feature. According to the discussion in Section 4.1, single feature is usually hard to meet various tracking challenges, thus fusing multiple features is necessary. Denote L different features as $\{\mathbf{F}_k : \mathbf{x} \rightarrow \Psi_k\}_{k=1}^L$, where $\mathbf{F}_k$ is a feature extraction function that projects an image patch $\mathbf{x}$ to its feature space Ψ_k . Motivated by [169] which proposes to re-determine the weight of each training sample to capture their importance more accurately, we propose a reliability re-determinative correlation filter (RRCF) to adaptively adjust the reliable score for each feature. It is formulated in the Fourier domain as

$$\begin{aligned} E(\hat{\mathbf{h}}_t, \mathbf{w}_t) &= \frac{1}{2} \sum_{k=1}^L w_{t,k} \|\hat{\mathbf{g}} - \hat{\Psi}_{t,k} \odot \hat{\mathbf{h}}_{t,k}\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^L \|\hat{\mathbf{h}}_{t,k}\|_2^2, \\ s.t. \quad &\sum_{k=1}^L w_{t,k} = 1, \quad w_{t,k} \geq 0, \end{aligned} \quad (4.2)$$

where $\hat{\mathbf{h}}_t$ is a concatenation of $\hat{\mathbf{h}}_{t,k}$, $\mathbf{w}_t$ is a $L \times L$ diagonal matrix containing the reliable score $w_{t,k}$ of each feature at the t -th tracking frame, a hat $\hat{\cdot}$ denotes the discrete Fourier transform of a vector.

In this paper, we attempt to jointly estimate $\hat{\mathbf{h}}_t$ and $\mathbf{w}_t$ by firstly fixing $\mathbf{w}_t$ to solve $\hat{\mathbf{h}}_t$ and then updating $\mathbf{w}_t$ given the estimated $\hat{\mathbf{h}}_t$.

4.2.1 Estimation of the CFs

When $\mathbf{w}_t$ is fixed, (4.2) has a closed-form solution for $\hat{\mathbf{h}}_{t,k}$, which is given as

$$\hat{\mathbf{h}}_{t,k} = \frac{w_{t,k} \hat{\Psi}_{t,k}^H \odot \hat{\mathbf{g}}}{w_{t,k} \hat{\Psi}_{t,k}^H \odot \hat{\Psi}_{t,k} + \lambda}, \quad (4.3)$$

where $\odot$ is Hadamard product operator, $\hat{\Psi}$ and Ψ^H denote the Fourier transform and complex-conjugate of Ψ , respectively.

From (4.3), it is clear that $\hat{\mathbf{h}}_{t,k}$ is an individual correlation filter (i.e. a sub-tracker) over the k -th feature type when $w_{t,k}$ is known. Then the response map of the k -th CF is computed as $\mathbf{R}_{t,k} = \mathcal{F}^{-1} \left(\hat{\mathbf{h}}_{t,k} \odot \hat{\Psi}_{t,k} \right)$, $\mathbf{R}_{t,k} \in \mathbb{R}^{W \times H}$. Thus, the translation $\mathbf{T}_{t,k} = [m_k, n_k]^T$ estimated based on the k -th feature at the t -th frame is denoted as

$$\mathbf{T}_{t,k} = \arg \max_{m_i, n_i} \mathbf{R}_{t,k}, \quad (4.4)$$

The final tracking result at t -th frame is a weighted average upon all the sub-trackers' translation estimates.

$$\mathbf{T}_t = \frac{1}{\sum w_{t,k}} \sum_{k=1}^L w_{t,k} \mathbf{T}_{t,k}. \quad (4.5)$$

The estimation for scale variation $\mathbf{S}_t$ of the target follows [38].

The updating for CF models is necessary to robustify the tracker in handling various environmental noise. In this paper, the final predicted states of the target will be applied to update all the CFs. Similar to the existing correlation filter tracking scheme [38, 170], the model updating is taking place by computing the numerator $\hat{\Pi}_{t,k}$ and denominator $\hat{\Theta}_{t,k}$ of (4.3) separately. An incremental manner is applied as follow

$$\hat{\Pi}_{t,k} = (1 - \gamma) \hat{\Pi}_{t-1,k} + \gamma \hat{\Psi}_{t,k}^H \odot \hat{\mathbf{g}}, \quad (4.6)$$

$$\hat{\Theta}_{t,k} = (1 - \gamma) \hat{\Theta}_{t-1,k} + \gamma \hat{\Psi}_{t,k}^H \odot \hat{\Psi}_{t,k}, \quad (4.7)$$

where $\gamma \in (0, 1)$ is a predefined learning rate. Meanwhile, a flowchart of the proposed tracking system is given in Fig. 4.1.

FIGURE 4.1: A framework of the proposed trackers. A CF response map is generated using each single feature, which is then fed to the proposed weight solver to find proper weights. The evaluated weights are applied to estimate the target's position and update the CF models.

4.2.2 Estimation of the Reliability $w_{t,k}$

In this section, to properly estimate $w_{t,k}$ given $\hat{\mathbf{h}}_{1:t,k}$, we propose two different weight solvers, named numerical optimization and model evaluation, respectively.

4.2.2.1 Estimating $w_{t,k}$ through Numerical Optimization

In this solution, to ensure the smoothness of $w_{t,k}$, an additional constraint is added on $w_{t,k}$, and reformulates (4.2) to

$$\arg \min_{\hat{\mathbf{h}}_t, \mathbf{w}_t} \frac{1}{2} \sum_{k=1}^L w_{t,k} \|\hat{\mathbf{g}} - \hat{\Psi}_{t,k} \odot \hat{\mathbf{h}}_{t,k}\|_2^2 + \frac{\lambda}{2} \sum_{k=1}^L \|\hat{\mathbf{h}}_{t,k}\|_2^2, \quad (4.8)$$

$$s.t. \quad \sum_{k=1}^L w_{t,k} = 1, \quad w_{t,k} \geq 0, \quad (4.9)$$

$$\|\mathbf{w}_t - \boldsymbol{\mu}_{t-\zeta:t}\|_2^2 \leq \varepsilon, \quad (4.9)$$

where $\boldsymbol{\mu}_{t-\zeta:t}$ denotes the mean value of $\mathbf{w}_t$ in the past ζ frames, ε is a pre-specified constant to limit the margin of $\mathbf{w}_t$.

Note that (4.8) is an inequality constrained optimization problem. To solve it, we introduce an interior point (IP) method to iteratively find out the optimal $\mathbf{w}_t$ given $\hat{\mathbf{h}}_t$. We first convert the inequality constraint (4.9) to an equality constraint

by introducing a new non-negative variable δ ,

$$\begin{aligned} s.t. \quad & \sum_{k=1}^L w_{t,k} - 1 = 0, \quad w_{t,k} \geq 0, \\ & \varepsilon - \delta - \|\mathbf{w}_t - \boldsymbol{\mu}_{t-\zeta:t}\|_2^2 = 0, \quad \delta \geq 0, \end{aligned} \quad (4.10)$$

Furthermore, the inequality constraints (i.e. $w_{t,k} \geq 0, \delta \geq 0$) can be removed by introducing the following penalty function

$$\mathbf{I}^+(x) = \begin{cases} 0, & x \geq 0, \\ \infty, & \text{otherwise.} \end{cases} \quad (4.11)$$

Denote CF's model loss of the k -th feature as e_t , i.e., $e_{t,k} = \|\hat{\mathbf{g}} - \hat{\boldsymbol{\Psi}}_{t,k} \odot \hat{\mathbf{h}}_{t,k}\|_2^2$. Then (4.8) is simplified as

$$\begin{aligned} \arg \min_{\mathbf{w}_t, \delta} & \frac{1}{2} \sum_{k=1}^L (w_{t,k} e_{t,k} + 2\mathbf{I}^+(w_{t,k})) + \mathbf{I}^+(\delta), \\ s.t. \quad & \sum_{k=1}^L w_{t,k} - 1 = 0, \\ & \varepsilon - \delta - \|\mathbf{w}_t - \boldsymbol{\mu}_{t-\zeta:t}\|_2^2 = 0, \end{aligned} \quad (4.12)$$

From [171], $\mathbf{I}^+(x)$ is approximated via a logarithmic barrier as $\mathbf{I}^+(x) \approx -\beta \log(x)$ where β is used to determine the approximation accuracy, i.e., $\beta = 1$. Note that a bigger β corresponds to a more accurate approximation. This is reasonable when the optimization begins at a positive initial value (i.e. $\delta_0 > 0$), since the logarithmic barrier will suppress variable δ towards zero and such suppression will be even stronger when $\delta \rightarrow 0$. In this way, the inequality constraints $w_{t,k} \geq 0$ and $\delta \geq 0$ can be ensured. With the Lagrange multiplier approach and logarithmic

barrier approximation, (4.12) is transformed to

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} \sum_{k=1}^L (w_{t,k} e_{t,k} - 2\beta \log(w_{t,k})) - \beta \log(\delta) + \\ & \lambda_w \left(\sum_{k=1}^L w_{t,k} - 1 \right) + \lambda_\delta (\varepsilon - \delta - \|\mathbf{w}_t - \boldsymbol{\mu}_{t-\zeta:t}\|_2^2), \end{aligned} \quad (4.13)$$

where λ_w and λ_δ denote the Lagrange multipliers.

Differentiating (4.13) w.r.t. $\mathbf{w}_t, \delta, \lambda_w, \lambda_\delta$ and applying the Newton iterative method, we obtain the optimal solution of $\mathbf{w}_t, \delta, \lambda_w, \lambda_\delta$ as

$$\begin{bmatrix} \Delta \mathbf{w}_t \\ \Delta \lambda_w \mathbf{I} \\ \Delta \lambda_\delta \mathbf{I} \\ \Delta \delta \mathbf{J}^T \end{bmatrix} = - \begin{bmatrix} \mathbf{H} & \mathbf{Q} & \mathbf{K} & 0 \\ \mathbf{I} & 0 & 0 & 0 \\ \mathbf{G} & 0 & 0 & \mathbf{J} \\ 0 & 0 & \lambda_\delta \mathbf{J}^T & \delta \end{bmatrix}^{-1} \begin{bmatrix} \partial \mathcal{L}_{\mathbf{w}_t} \mathbf{w}_t \\ \partial \mathcal{L}_{\lambda_w} \mathbf{I} \\ \partial \mathcal{L}_{\lambda_\delta} \mathbf{I} \\ \delta \partial \mathcal{L}_\delta \mathbf{J}^T \end{bmatrix} \quad (4.14)$$

where $\mathbf{H} = \frac{1}{2} e_t + 4\lambda_\delta \mathbf{w}_t - 2\lambda_\delta \boldsymbol{\mu}_{t-l:t} + \lambda_w$, $\mathbf{G} = -2(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t})$, $\mathbf{K} = 2(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t}) \mathbf{w}_t$, $\mathbf{Q} = \mathbf{w}_t$, $\mathbf{I} \in \mathbb{R}^{L \times L}$ denotes the identity matrix with size L and $\mathbf{J} \in \mathbb{R}^{L \times 1}$ denotes a vector with ones. More details for the deviation from (4.13) to (4.14) is presented in (9.1).

Thus, with an estimate $(\mathbf{w}_t^j, \delta^j, \lambda_w^j, \lambda_\delta^j)$ at the j -th step, under the interior of the bound constraints (i.e. $w_{t,k}, \delta \geq 0$), the estimate at the next step is updated by

$$\begin{aligned} (\mathbf{w}_t^{j+1}, \delta^{j+1}, \lambda_w^{j+1}, \lambda_\delta^{j+1}) = & (\mathbf{w}_t^j, \delta^j, \lambda_w^j, \lambda_\delta^j) + \\ & (\gamma_w \Delta \mathbf{w}_t, \gamma_\delta \Delta \delta, \gamma_{\lambda_w} \Delta \lambda_w, \gamma_{\lambda_\delta} \Delta \lambda_\delta), \end{aligned} \quad (4.15)$$

where $\gamma_w, \gamma_\delta, \gamma_{\lambda_w}$ and γ_{λ_δ} are the updating step for $\mathbf{w}_t, \delta, \lambda_w$ and λ_δ , respectively. They are all set to 0.05 in the evaluation experiments.

4.2.2.2 Estimating $w_{t,k}$ through Model Evaluation

The method presented in Section 4.2.2.1 has a drawback that the optimization highly depends on the instantaneously evaluated model loss $e_{t,k}$ which cannot properly differentiate the target apart from its surrounding background in the presence of the environmental noise, such as occlusion and background clutter. Therefore, we propose to learn an independent discriminative model to evaluate $w_{t,k}$ by preserving some important historical appearances of the target. The general process of model evaluation based tracker is summarized as the following four steps: 1) estimating the translation $\mathbf{T}_{t,k}$ and scale variation $\mathbf{S}_{t,k}$ of the target using single feature based CF, 2) re-determining the weight for each feature using the learned SVR model at the previous frame, 3) computing the bounding box $\mathbf{B}_t$ of the target based on (4.5), 4) selecting the samples at the surrounding of $\mathbf{B}_t$ to update the SVR model.

Intuitively, $w_{t,k}$ is inversely proportional to the drift occurring on its corresponding sub-tracker. A machine learning approach, i.e. support vector regression (SVR), is applied to evaluate such tracking drift. Denote the bounding box of the target as $\mathbf{B}_t = [\mathbf{T}_t^T, \mathbf{S}_t^T]^T$, where $\mathbf{S}_t = [w_t, h_t]^T$ is the size of $\mathbf{B}_t$. At the t -th tracking frame, samples $\{\mathbf{x}_i\}_{i=1}^M$ at the surrounding of $\mathbf{B}_t$ are selected to incrementally train and update the tracking drift evaluation model, where M is the number of samples. To best present the connection between the training sample and its possible drift compared to the "true" target, a label y_i is assigned to each sample as $y_i = \frac{\mathbf{B}_i \cap \mathbf{B}_t}{\mathbf{B}_i \cup \mathbf{B}_t}$, where $\mathbf{B}_i$ denotes the state of the i -th sample.

Typically, the objective of SVR model is to find a function $\psi(\mathbf{x}) = \langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle + b$ which can predict the label for new samples. With the training samples $\{\mathbf{x}_i\}_{i=1}^M$ and their corresponding labels $\{y_i\}_{i=1}^M$, $\psi(\mathbf{x})$ can be found by solving the following

convex optimization problem

$$\begin{aligned}
& \arg \min_{\boldsymbol{\theta}, \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^*} \quad \frac{1}{2} \|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^M (\boldsymbol{\xi}_i + \boldsymbol{\xi}_i^*), \\
& s.t. \quad y_i - \langle \boldsymbol{\theta}, \phi(\mathbf{x}_i) \rangle - b \leq \epsilon + \boldsymbol{\xi}_i, \\
& \quad \langle \boldsymbol{\theta}, \phi(\mathbf{x}_i) \rangle + b - y_i \leq \epsilon + \boldsymbol{\xi}_i^*, \\
& \quad \boldsymbol{\xi}_i, \boldsymbol{\xi}_i^* \geq 0, \quad i = 1, \dots, M
\end{aligned} \tag{4.16}$$

where $\phi(\mathbf{x})$ is the kernel function and $\langle \cdot, \cdot \rangle$ denotes the dot product operator. The positive constant C determines the trade-off between the flatness of ψ and the amount up to which deviations larger than a pre-specified ϵ , which is set to 0.1.

Using the notion of duality, (4.16) can be converted into its equivalent dual form as

$$\begin{aligned}
& \arg \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M K_{ij} (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \\
& \quad - \sum_{i=1}^M y_i (\alpha_i - \alpha_i^*) + \sum_{i=1}^M \epsilon (\alpha_i + \alpha_i^*), \\
& s.t. \quad \sum_{i=1}^M (\alpha_i - \alpha_i^*) = 0, \\
& \quad \alpha_i, \alpha_i^* \in [0, C], i = 1, \dots, M.
\end{aligned} \tag{4.17}$$

where K is a *kernel matrix* and contains the values of kernel function $K_{i,j}$ defined as $K_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_j), \phi(\mathbf{x}_i) \rangle$. Thus, the evaluation function ψ is written in its dual form as

$$\psi(\mathbf{x}_i) = \sum_{j=1}^M (\alpha_j - \alpha_j^*) K_{i,j} + b. \tag{4.18}$$

Usually, the evaluation model should be updated incrementally to make the learned model be consecutive to the target's appearance variation. Following the incrementally updating scheme of SVR proposed in [172] and [173], we denote $\tau = \alpha - \alpha^*$,

and define the margin function $h(\mathbf{x}_i)$ as

$$h(\mathbf{x}_i) = \psi(\mathbf{x}_i) - y_i = \sum_{j=1}^M K_{ij}\tau_j - y_i + b. \quad (4.19)$$

According to KKT conditions, we can find the support vectors as $\mathcal{S} = \{i | 0 < \tau_i < C\}$.

Whenever a new sample $\mathbf{x}_c$ is added to the training set, our goal is to find its corresponding weight τ_c as well as adjust the weight τ_i in support set $\mathcal{S}$ to ensure that all of them meet the KKT conditions. Denote the variants of b and $\boldsymbol{\tau}$ as Δb and $\Delta \boldsymbol{\tau}$, respectively, the incremental relation between $\Delta h(\mathbf{x}_i)$, $\Delta \tau_i$ and Δb is given as:

$$\begin{cases} \sum_{j \in \mathcal{S}} \Delta \tau_j = -\Delta \tau_c \\ \sum_{j \in \mathcal{S}} K_{ij} \Delta \tau_j + \Delta b = -K_{ic} \Delta \tau_c, \quad i \in \mathcal{S} \end{cases} \quad (4.20)$$

which can be expressed in the matrix operation as:

$$\begin{bmatrix} \Delta b \\ \Delta \tau_{s_1} \\ \vdots \\ \Delta \tau_{s_l} \end{bmatrix} = - \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{s_1 s_1} & \cdots & K_{s_1 s_l} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{s_l s_1} & \cdots & K_{s_l s_l} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ K_{s_1 c} \\ \vdots \\ K_{s_l c} \end{bmatrix} \Delta \tau_c, \quad (4.21)$$

where l is the number of support vectors. More details about the online learning of SVR model are provided in Section 9.2.

Finally, according to (4.18), the drift on each sub-tracker is obtained as $\{\psi(\mathbf{x}_{t,i})\}_{i=1}^L$, which denotes the state of the target estimated according to each feature. It is clear that $\mathbf{x}_{t,i} \rightarrow 1$ means a slight drift, while $\mathbf{x}_{t,i} \rightarrow 0$ denotes a great one. Therefore, $w_{t,k}$ is computed as

$$w_{t,k} = w_{t-1,k} \cdot \exp \left(-\sigma_a (\psi(\mathbf{x}_{t,k}) - 1)^2 \right), \quad (4.22)$$

where σ_a is a predefined value to shape the distribution of estimation. In the evaluation experiments, σ_a is set to 0.3.

TABLE 4.1: Tracking performance evaluation with different individual feature and their combination. **Red**: the best. **Blue**: the second best.

	CN	HOG	<i>conv</i> 4-4	<i>conv</i> 5-4	<i>conv</i> 4-4 <i>conv</i> 5-4	<i>conv</i> 4-4 <i>conv</i> 5-4 HOG+CN	RRCF _{NO/H}	RRCF _{NO}
OS (AUC)	51.8	56.9	62.7	63.3	65.4	66.3	64.3	69.3
DP (20px)	66.6	74.7	82.6	85.4	86.6	87.4	84.9	91.2

TABLE 4.2: Evaluating the impact of adaptively adjusting the regularization penalty of CF. The scores listed in the table denote OP (@AUC) and DP (@20px), respectively.

	OTB-100		TempleColor	
	OP (%)	DP (%)	OP (%)	DP (%)
RRCF _{NW}	67.2	89.3	55.6	77.3
RRCF _{NO}	69.3	91.2	59.3	80.7
RRCF _{ME}	69.9	92.0	59.2	80.1

Finally, according to (4.18), the drift on each sub-tracker is obtained as $\{\psi(\mathbf{x}_{t,i})\}_{i=1}^L$, where $\{\mathbf{x}_{t,i}\}_{i=1}^L$ are the objects estimated by each sub-tracker. It is clear that $\mathbf{x}_{t,i} \rightarrow 1$ means a slight drift, while $\mathbf{x}_{t,i} \rightarrow 0$ denotes a great drift. Therefore, $w_{t,k}$ is computed as

$$w_{t,k} = w_{t-1,k} \cdot \exp(-\sigma_a(\psi(\mathbf{x}_{t,k}) - 1)^2), \quad (4.23)$$

where σ_a is a predefined value.

4.3 Experimental Tests and Results

4.3.1 Implementation Details

The proposed tracking framework is implemented in two different trackers. We denote the trackers using numerical optimization and model evaluation to estimate the feature’s reliable score as RRCF_{NO} and RRCF_{ME} , respectively. The features applied in our tracking approaches include HOG, CN, conv4-4 and conv5-4 of VGG-19 [174]. Meanwhile, both conv4-4 and conv5-4 are divided into three independent features to enlarge the feature pools. For example, conv4-4 is divided into the lower 256, the higher 256 channels of conv4-4 and the whole channel of conv4-4 . Note that the deep feature is directly fed to a correlation filter to compute its corresponding response map, similar to the way used for handcraft feature.

The evaluation datasets are OTB-100 [2], TempleColor [33], VOT2016 [3], VOT2018 [81] and LaSOT [168]. For fair comparisons, we employ the publicly available codes or results provided by authors in all the relevant comparisons. Meanwhile, both RRCF_{NO} and RRCF_{ME} are using the same feature configuration and their parameters keep the same through all the evaluations.

4.3.2 Tracking Framework Study

4.3.2.1 Tracking with different features

We evaluate the tracking performance for each individual feature on OTB-100, including HOG, CN, conv4-4 of VGG-19, conv5-4 of VGG-19 and some of their combinations. The tracker we apply for this comparison is RRCF_{NO} . Meanwhile, $\text{RRCF}_{\text{NO/H}}$ denotes the tracker only applies handcraft features (i.e. HOG and CN). The detail experimental results are listed in Table 4.1, from which, we know that deep features have great improvements on the tracking accuracy compared to handcraft features. An even greater improvement in terms of distance precision (DP)

TABLE 4.3: Comparisons of our approach with the state-of-the-art trackers under distance precision (@20px) on OTB-100 and TempleColor. **Red**: the best. **Blue**: the second best.

	when	where	Need GPU	OTB-50(%)↑	OTB-100(%)↑	TempleColor(%)↑
CCOT	2016	ECCV	Y	89.0	89.3	78.0
DeepSRDCF	2015	ICCVW	Y	83.1	84.2	73.7
ECO	2017	CVPR	Y	91.2	90.1	79.7
SiamDW	2019	CVPR	Y	91.5	91.4	-
MCPF	2017	CVPR	Y	90.2	86.6	77.3
BACF	2017	ICCV	N	86.9	82.7	65.2
Staple_CA	2017	CVPR	N	83.2	80.9	67.9
PTAV	2017	ICCV	Y	89.4	84.7	73.8
MCCT	2018	CVPR	Y	92.8	91.7	79.3
TRACA	2018	CVPR	Y	88.2	80.6	-
CSRDCF	2017	CVPR	N	78.7	79.3	-
VITAL	2018	CVPR	Y	93.4	90.9	78.8
DSLT	2018	CVPR	Y	91.9	90.0	80.7
TADT	2019	CVPR	Y	88.1	85.7	75.6
UDP+	2019	CVPR	Y	83.0	83.0	71.7
SiamRPN++	2019	CVPR	Y	90.2	90.6	-
RRCF _{NO}	2019	this work	Y	92.1	91.2	80.7
RRCF _{ME}	2019	this work	Y	93.2	92.0	80.1

and overlap precision (OP) are achieved when concatenating four features as one integrated feature. Meanwhile, when comparing the proposed tracker RRCF_{NO/H} to single deep feature based trackers, RRCF_{NO/H} achieves the performance gain of 1.6% and 2.3% in terms of DP and OP, respectively. According to this experiment, it is clear that 1) single feature based tracker is hard to achieve a satisfactory tracking accuracy, 2) a simple combination of multiple features only improves a small gap of accuracy. Therefore, it is necessary to design an effective fusion framework to take the advantage of multiple features properly.

4.3.2.2 Evaluation on the effects of w_t

In this experiment, we attempt to evaluate the effectiveness of the proposed reliability re-determinative scheme. The comparison trackers include a tracker which is

not to adjust $\mathbf{w}_t$ and fix $\mathbf{w}_t$ to $1/L$, which is labelled this tracker as RRCF_{NW} , and the proposed RRCF_{NO} and RRCF_{ME} . The evaluation is conducted on OTB-100 and TempleColor. Based on the comparison results shown in Table 4.2, we know that the proposed adaptive re-determination on the reliability for each feature can bring a significant improvement upon the baseline tracker (i.e. RRCF_{NW}). In detail, RRCF_{NO} averagely improves 2.9% and 2.7%, while RRCF_{ME} improves 3.2% and 2.8% in terms of OP and DP on OTB-100 and TempleColor, respectively.

FIGURE 4.2: Accuracy of our approach with different values of ϵ and C .

4.3.2.3 Parameter selection

In this section, we investigate the performances of the proposed two weight solvers with different choices of parameters. For the parameters of each CF tracker, like γ and λ , we follow [38] to set. We focus on the effects of key parameters in the proposed weight solvers, namely ϵ in (4.9) and C in (4.17). While ϵ determines the smoothness of the evaluated weights, C controls the trade-off between achieving a low error on the training data and minimising the norm of the weights. By selecting different values of ϵ and C , we evaluate the accuracy of the proposed tracker on OTB-100. The obtained results are shown in Fig. 4.2, according to which the best performances are observed when ϵ is 0.1 and C is 1. These values will be used in the following evaluation experiments.

FIGURE 4.3: Quantitative results on the OTB-50 dataset with 51 videos. The scores reported in the legend of the left and right figures are precision score at 20px and average AUC values, respectively.

FIGURE 4.4: Quantitative results on the OTB-100 dataset with 100 videos. The scores reported in the legend of the left and right figures are precision scores at 20px and average AUC values, respectively.

4.3.3 Comparison with state-of-the-art trackers

4.3.3.1 Evaluation on OTB-50 and OTB-100

OTB-50 [1] and OTB-100 [2] are two popular used datasets, where OTB-50 contains 51 videos with various targets, and OTB-100 is expanded from OTB-50 which contains 100 videos. In this comparison, we evaluate the proposed trackers RRCF_{NO} and RRCF_{ME} with 16 recent state-of-the-art trackers, including MCPF

FIGURE 4.5: Quantitative results on the TempleColor dataset with 129 videos. The scores reported in the legend of the left and right figures are precision scores at 20px and average AUC values, respectively.

FIGURE 4.6: Expected overlap curves plots. The score reported in the legend is the EAO score which takes the mean of expected overlap between two purple dot vertical lines.

[117], MCCT [55], ECO [50], CCOT [98], SiamRPN++ [51], DeepSRDCF [89], VITAL [175], DSLT [176], PTAV [177], TRACA [111], TADT [178], UDP+ [179], BACF [92], CSRDCE [93] Staple_CA [87] and SiamDW [180] on these two datasets.

We illustrate the results in one-pass evaluation and only top 14 trackers are shown in Fig. 4.3 and Fig. 4.4. On OTB-50 dataset, the proposed RRCF_{ME} achieves the best and the second best tracking accuracies in terms of OP and DP, respectively.

FIGURE 4.7: Expected overlap curves plots. The score reported in the legend is the EAO score which takes the mean of expected overlap between two purple dot vertical lines.

In detail, RRCF_{ME} outperforms the second best tracker VITAL 0.9% in terms of OP, while VITAL only outperforms RRCF_{ME} 0.2% in terms of DP.

On OTB-100 dataset, the proposed RRCF_{ME} achieves the best and second best tracking accuracies in terms of DP and OP, respectively. In detail, SiamRPN++ exceeds the proposed tracker 0.2% in terms of OP, while RRCF_{ME} outperforms it 1.4% on DP. Furthermore, the proposed RRCF_{ME} outperforms SiamRPN++ 3.0% and 2.9% in terms of DP and OP, respectively, on OTB-50 dataset. Overall, according to the comparisons on these two datasets, we can conclude that the proposed RRCF_{ME} achieves the top tracking performance on OTB-50 and OTB-100.

Furthermore, another proposed tracker RRCF_{NO} also achieves an outstanding performance, such as on OTB-100, RRCF_{NO} outperforms VITAL which achieves the best and the second best on OTB-50. While on OTB-50, RRCF_{NO} outperforms SiamRPN++ which achieves the best and the second best on OTB-100.

TABLE 4.4: Comparisons to the state-of-the-art trackers on the VOT2016. The results are presented in terms of EAO, accuracy and failure. **Red**: the best. **Blue**: the second best. **Green**: the third best.

	VOT2016			VOT2018		
	Accuracy $\uparrow$	Failure $\downarrow$	EAO $\uparrow$	Accuracy $\uparrow$	Failure $\downarrow$	EAO $\uparrow$
SiamMASK	0.61	0.77	0.430	0.60	1.00	0.384
DeepSRDCF	0.51	1.17	0.276	0.484	2.51	0.154
ECO	0.54	0.72	0.374	0.476	0.98	0.281
CSRDCF	0.51	0.85	0.335	0.485	1.27	0.256
CCOT	0.52	0.85	0.331	0.485	1.13	0.267
MCCT	0.57	0.67	0.395	0.526	1.13	0.278
DAT	0.55	0.97	0.320	-	-	-
VITAL	0.55	0.98	0.323	-	-	-
MDNet_N	0.53	1.20	0.257	-	-	-
UDP+	0.50	1.10	0.302	-	-	-
TADT	0.54	1.17	0.300	-	-	-
DaSiamRPN	0.56	1.12	0.340	-	-	-
DLST	0.53	0.78	0.345	-	-	-
DLSTpp	-	-	-	0.542	0.80	0.325
DSiam	-	-	-	0.506	2.37	0.195
SiamDW	-	-	-	0.525	1.42	0.273
TRACA	-	-	-	0.431	3.05	0.139
SiamFC	-	-	-	0.498	2.08	0.187
DeepCSRDCF	-	-	-	0.483	0.98	0.293
RRCF _{NO}	0.56	0.58	0.428	0.527	1.05	0.291
RRCF _{ME}	0.57	0.53	0.453	0.538	0.89	0.335

4.3.3.2 Evaluation on TempleColor

We evaluate the proposed tracker with 12 recent state-of-the-art trackers, including MCPF [117], MCCT [55], ECO [50], CCOT [98], VITAL [175], DSLT [176], PTAV [177], TADT [178], UDP+ [179], DeepSRDCF [89], BACF [92] and Staple_CA [87] on TempleColor [33], which contains 128 video sequences with various targets. The evaluation results are shown in Fig. 4.5. The proposed tracker RRCF_{NO}

FIGURE 4.8: Quantitative results on OTB-100 for 8 challenging attributes: background clutters, deformation, fast motion, in-plane rotation, illumination variation, motion blur, occlusion and out-of-plane rotation.

achieves the best and second best in terms of DP and OP. Though ECO outperforms RRCF_{NO} 1.1% in terms of OP, it is exceeded 2.0% by RRCF_{NO} in terms of DP. More importantly, the proposed RRCF_{ME} also achieves the second best and third best in terms of DP and OP.

When comparing on the above mentioned three datasets, both of the proposed RRCF_{ME} and RRCF_{NO} achieve a steady yet accurate tracking results, which demonstrates that 1) the proposed adaptive re-determination on the reliability of each feature can actually improve the tracking accuracy, 2) the proposed approaches are more suitable for various/general challenging sequences (e.g. both perform well on OTB-50, OTB-100 and TempleColor) than other state-of-the-art trackers.

4.3.3.3 Evaluation on LaSOT

We evaluate the proposed trackers by comparing them with 9 recent state-of-the-art trackers, including DSiam [103], STRCF [53], ECO [50], CFNet [110], BACF [92], TRACA [111], PTAV [177], Staple_CA [87], and CSRDCF [93] on LaSOT [168]. LaSOT contains 280 video sequences with average 3000 frames per sequence and thus it is a challenging dataset that is a good choice to evaluate the long-term tracking ability. The evaluation results are shown in Fig. 4.9. The proposed RRCF_{ME} and RRCF_{NO} achieve the best and the third best in terms of DP and

OP, respectively. Although the tracking accuracy on LaSOT is not so favourable compared to the results achieved on OTB-100 and TempleColor, the proposed two trackers still achieve important improvements compared to other short-term trackers, such as ECO which performs well on TempleColor and OTB-100. It demonstrates that the proposed trackers have the ability to alleviate the issue of model drift, thus can work better on the long-term sequences.

FIGURE 4.9: Precision and success plots on the LaSOT dataset with 280 videos. The scores reported in the legend of the left and right figures are precision scores at 20px and average AUC values, respectively.

4.3.3.4 Evaluation on VOT2016

Furthermore, the proposed approaches are compared with 13 recent state-of-the-art trackers, including MCCT [55], CSRDCF [93], ECO [50], DeepSRDCF [89], VITAL [175], CCOT [98], DaSiamRPN [109], UDP+ [179], TADT [178], DSLT [176], DAT [96], MDNet_N [95] and SiamMask [181] on VOT2016 [3]. The primary of VOT2016 measure focuses on the expected average overlap (EAO) on the testing sequences. The EAO is actually the expected no-reset average overlap, which is similar to AUC score in OTB methodology, but with reduced bias and variance as explained in [3].

Table 4.4 presents the comparison results of 12 state-of-the-art trackers under the metric of EAO, accuracy and robustness. Fig. 4.6 shows the expected overlap

curves of the comparison trackers. From these results we can observe: 1) The proposed RRCF_{ME} and RRCF_{NO} achieve the best EAO (e.g. 0.453) and third best EAO (e.g. 0.428) which all outperform the winner of VOT2016 challenge (CCOT) a large margin. 2) Both of the proposed RRCF_{ME} and RRCF_{NO} achieve an outstanding performance on the failure score (e.g. the best and second best, respectively). 3) From the expected overlap curves shown in Fig. 4.6, it is clear to see that RRCF_{ME} outperforms the remaining trackers a large margin along all the sequence length. Overall, the experimental results significantly demonstrate that the proposed reliability re-determination approaches are effective on alleviating the issue of model drift.

4.3.3.5 Evaluation on VOT2018

Finally, the proposed approaches are compared with 12 recent state-of-the-art trackers, including MCCT [55], SiamDW [180], ECO [50], CSRDCF [93], DeepCSRDCF [93], DSiam [103], CCOT [98], SiamFC [102], TRACA [111], DSLTpp [176], DeepSRDCF [89] and SiamMask [181] on VOT2018 [81]. The primary of VOT2018 measure is similar to VOT2016 which focuses on EAO on the testing sequences.

Table 4.4 presents the comparison results of 12 state-of-the-art trackers under the metric of EAO, accuracy and robustness. Fig. 4.7 shows the expected overlap curves of the comparison trackers. The proposed RRCF_{ME} achieves the second best EAO score (e.g. 0.335), at the same time, RRCF_{NO} also obtains a competitive tracking results in terms of EAO score (e.g. 0.291)

From the comparison results across the VOT2016 and VOT2018 datasets, we can observe that the proposed RRCF_{ME} and RRCF_{NO} achieve a stable and also accurate tracking results. For examples RRCF_{ME} and RRCF_{NO} have an impressive EAO and failure score on VOT2016 which significantly outperform the remaining trackers, while on VOT2018, both the proposed two trackers still achieve a competitive score in terms of EAO and failure score. While the trackers MCCT and ECO

achieve favourable EAO scores on VOT2016, they cannot continuously achieve such high EAO scores in VOT2018. The outstanding tracking results on VOT2016 and VOT2018 datasets further valid the effectiveness of the proposed methods.

4.3.4 Comparison between the proposed two solutions

We evaluate the effectiveness of the proposed two solutions about how to find the reliable score for different features. The experimental results on five datasets are given in Table 4.3 and Table 4.4. It is clear that RRCF_{ME} assesses the reliability of different features more accurately, thus it achieves a better tracking accuracy than RRCF_{NO} . The most important reason is that RRCF_{ME} learns a SVR model to evaluate the weights, while RRCF_{NO} evaluates the weights using the loss computed from CF model, i.e., $e_{t,k} = \|\hat{\mathbf{g}} - \hat{\Psi}_{t,k} \odot \hat{\mathbf{h}}_{t,k}\|_2^2$, at each tracking frame. Since RRCF_{ME} maintains a pool of support samples which preserve the target’s historical appearance information, it can provide a stable estimation on the weights in the presence of occlusion and background clutter. As demonstrated in Fig. 4.8, RRCF_{ME} has achieved a much better accuracy on the challenging attributes: occlusion, background clutter and illumination variation. However, with the improvement on the tracking accuracy, RRCF_{ME} requires more computational resource due to the online learning of the SVR model.

4.3.5 Comparisons on different attributes on OTB-100

To thoroughly evaluate the robustness of the proposed two trackers in various scenes, we present tracking accuracy in terms of 8 challenging attributes on OTB-100 as shown in Fig. 4.8. As observed from the figure, the proposed two trackers can handle various tracking challenges steadily and accurately, while none of other trackers can achieve such a stable yet accurate tracking result. Particularly, when handling the attribute of occlusion, deformation, out-of-plane rotation, fast motion, background clutters and illumination variation, which can easily result in tracking

drift, the proposed trackers can address these challenges appropriately due to that the proposed tracking frameworks can properly take the advantages from different features.

Overall, extensive comparison results have demonstrated that the proposed two trackers can work robustly yet achieve top rank tracking performance among five most popular datasets, which contain various challenges sequences.

4.3.6 Qualitative evaluation

We illustrate the tracking results of the proposed two trackers and other five state-of-the-art trackers on the typical sequences shown in Fig. 4.10. These sequences contain some extremely challenging attributes, such as fast motion, occlusion, in-plane rotation, background clutter, *et al.* Overall, the proposed trackers can accurately carry out the tracking task on almost all sequences, while none of other trackers can achieve this. On *Messi_ce* (row 3, column 1), *MotorRolling* (row 2, column 2) and *Panda* (row 4, column 1), only the proposed RRCF_{ME} and MCPF successfully track the target, while the scale estimation of RRCF_{ME} is much better than MCPF . However, the proposed RRCF_{NO} fails on these three sequences due to that the instantaneous loss of each weaker tracker cannot truly present the weight of each feature when the target is heavily occluded or its appearance information varies greatly. Meanwhile, the tracking performances of MCPF in the remaining sequences are not so satisfactory, such as in *Airport_ce* (row 1, column 2), *Skating2-1* (row 3, column 2) and *Spiderman_ce* (row 5, column 2). Similarly, other trackers also fail on at least four sequences.

4.3.7 Analysis on the tracking speed

The proposed trackers are evaluated on a laptop with an Intel[®] Core i7 2.60GHz CPU and a Nvidia GeForce RTX 2060 GPU. With such a hardware platform, RRCF_{ME} and RRCF_{NO} achieve 3.2 and 9.8 FPS, respectively. To facilitate more

FIGURE 4.10: Qualitative tracking results with the proposed two trackers and other state-of-the-art trackers.

details on the computational cost of the proposed tracker, we evaluate the time consumption on three main steps: 1) feature extraction, 2) learning and updating for CFs, and 3) the proposed weight solvers. Table 4.5 presents the average time consumption for the above steps in OTB-100 dataset. It is clear seen that model evaluation costs more computational resource than numerical optimization, but generally achieves an accurate yet robust tracking results.

TABLE 4.5: Time evaluation on the steps of the proposed tracker.

	feature extraction	learn CFs	weight solver	total
RRCF_{ME}	32.6ms	52.7ms	226ms	312ms
RRCF_{NO}	33.1ms	53.8ms	15.1ms	102ms

4.3.8 Discussion on pros and cons

The proposed methods allow the weight of each feature to be adaptively adjusted, while the existing methods do not have this flexibility. Extensive comparison results have demonstrated that such flexibility is beneficial to enhance the tracking accuracy and robustify the tracking models. Due to the fact that different features are extracted under different point of view, the proposed adaptive adjustment on

the features' weight could enable the tracker to trust more on the more suitable features while suppress the others, thus the proposed tracker can work on more challenging sequences.

The target of the proposed tracking scheme is to take advantage of each feature, thus make the task of fusing multiple features not simply be " $1 + 1 = 2$ ", but " $1 + 1 > 2$ ". For example, according to (4.1), the proposed RRCF_{NO} improves 3% and 3.8% than the tracker directly integrated from the handcraft and deep features in terms of OP and DP, respectively. With more stronger features, most of the CF-based trackers can achieve better accuracies, while the proposed tracking scheme has the ability to achieve even more significant improvement.

In order to achieve such improvement, more computational resources are required. Thus, a proper balance between the accuracy and the tracking speed should be made properly for the proposed tracker. Note that the proposed approaches focus on adjusting the weights for each feature model, without paying attention to train more powerful deep models which can represent the target more stably in some extremely challenging scenarios such as abrupt motion blur and serious background clutter. These are future interesting research topics for investigation.

4.4 Conclusion

In this paper, we have proposed a novel tracking framework which adaptively re-determines the importance of each feature space. Meanwhile, two different evaluation models, named model evaluation and numerical optimization, are proposed and implemented to achieve this goal. Extensive experiments demonstrate that both of the proposed trackers perform favourably against state-of-the-art trackers.

Chapter 5

UWB/LiDAR Fusion SLAM via One-step Optimization

5.1 Introduction

In this section, we propose a 2D range-only SLAM approach that combines low-cost UWB sensors and LiDAR sensor for simultaneously localizing the robot, beacons and constructing a 2D map of an unknown environment in real-time. Here are some highlights of the proposed approach:

1. no prior knowledge about the robot's initial position is required, neither is control input;
2. beacon(s) may be moved and number of beacon(s) may be varied while SLAM is proceeding;
3. the robot can move fast without accumulating errors in constructing map;
4. the map can be built even in feature-less environment.

To facilitate the understanding of the proposed algorithm, we brief the symbols used in this chapter in Table [5.1](#).

TABLE 5.1: Notations for the symbols used in this chapter.

Symbols	Description
$\mathbb{R}$	The set of real numbers
superscript T	Matrix transpose
superscript $^{-1}$	Matrix inverse
$\mathbf{I}_n$	An identity matrix of size $n \times n$
$\mathbf{1}_n$	A column vector with n elements all being one
$\otimes$	Kronecker product
$ \cdot $	Absolute value
$\mathcal{N}(\mu, \sigma^2)$	The normal distribution with mean μ and variance σ^2
$[\mathbf{A}]_i$	The i th row of matrix $\mathbf{A}$
$[\mathbf{A}]_{i,j}$	The element at the i th row and the j th column of matrix $\mathbf{A}$

5.2 Problem Formulation

We consider to fuse two kinds of ranging measurements which are

- the peer-to-peer ranges between UWB nodes consisting of robot-to-beacon ranges and beacon-to-beacon ranges, and
- the range between the robot and obstacles available from laser range finder.

We denote UWB-based ranging measurements as $\mathbf{r}^{(\text{UWB})}$ and LiDAR-based ranging measurements as $\mathbf{r}^{(\text{LiD})}$. Let

$$\begin{aligned}\mathbf{x}_t &= [p_{0,x,t}, p_{0,y,t}, v_{0,x,t}, v_{0,y,t}]^T, \\ \mathbf{m}_t^{(\text{UWB})} &= [p_{1,x,t}, p_{1,y,t}, \dots, p_{N_t,x,t}, p_{N_t,y,t}, \\ &\quad v_{1,x,t}, v_{1,y,t}, \dots, v_{N_t,x,t}, v_{N_t,y,t}]^T,\end{aligned}$$

where $\mathbf{x}_t \in \mathbb{R}^4$ contains 2D location of the robot $\mathbf{p}_{0,t} = [p_{0,x,t}, p_{0,y,t}]^T$ and 2D velocity $\mathbf{v}_{0,t} = [v_{0,x,t}, v_{0,y,t}]^T$ at time t , $\mathbf{m}_t^{(\text{UWB})} \in \mathbb{R}^{4N_t}$ which represents "UWB map" contains N_t number of UWB beacons' 2D locations $\mathbf{p}_{n,t} = [p_{n,x,t}, p_{n,y,t}]^T$, $n =$

$1, \dots, N_t$ and 2D velocities $\mathbf{v}_{n,t} = [v_{n,x,t}, v_{n,y,t}]^T, n = 1, \dots, N_t$, and let $\mathbf{m}_t^{(\text{LiD})}$ be the "LiDAR map" containing the locations of the observed obstacles.

Remark 1. No control input e.g. odometry/IMU/etc., is assumed. The heading information (or equivalently 2D velocity) for robot/UWB beacons is derived purely from their trajectory.

Given all ranging measurements $\mathbf{r}_{1:t}^{(\text{UWB})}$ and $\mathbf{r}_{1:t}^{(\text{LiD})}$ up to time t , our goal is to simultaneously estimate $\mathbf{x}_t$, $\mathbf{m}_t^{(\text{UWB})}$ and $\mathbf{m}_t^{(\text{LiD})}$. We propose to decompose the problem into three coupled steps:

1. find the relative positions of robot and beacons based on $\mathbf{r}_{1:t}^{(\text{UWB})}$, and derive the robot's heading information from its trajectory,
2. construct/update LiDAR map as well as UWB map using $\mathbf{r}_{1:t}^{(\text{LiD})}$ and robot's pose and heading estimates and beacons' pose estimates,
3. correct robot's pose and heading as well as beacons' poses based on the map's feedback.

In what follows, we will elaborate how we implement these three steps.

5.3 UWB-Only SLAM

5.3.1 The dynamical and observational models

Let $\boldsymbol{\chi}_t = [\mathbf{p}_t^T, \mathbf{v}_t^T]^T \in \mathbb{R}^{4(N_t+1)}$ represent the complete UWB-related state consisting of locations of robot and beacons $\mathbf{p}_t = [\mathbf{p}_{0,t}^T, \dots, \mathbf{p}_{N_t,t}^T]^T$ and velocities of robot and beacons $\mathbf{v}_t = [\mathbf{v}_{0,t}^T, \dots, \mathbf{v}_{N_t,t}^T]^T$. The state $\boldsymbol{\chi}_t$ is evolved according to the following dynamical model:

$$\boldsymbol{\chi}_t = \mathbf{F}_t \boldsymbol{\chi}_{t-1} + \mathbf{G}_t \mathbf{w}_t,$$

where the transition matrix $\mathbf{F}_t$ equals to $\begin{bmatrix} \mathbf{I}_{2(N_t+1)} & \delta \mathbf{I}_{2(N_t+1)} \\ \mathbf{0} & \mathbf{I}_{2(N_t+1)} \end{bmatrix}$, and $\mathbf{G}_t$ equals to $\begin{bmatrix} \delta \mathbf{I}_{2(N_t+1)} \\ \mathbf{I}_{2(N_t+1)} \end{bmatrix}$, the state noise $\mathbf{w}_t$ is zero-mean and has covariance $\mathbf{Q}_t = \sigma_w^2 \mathbf{G}_t \mathbf{G}_t^T$ [182] and δ is the sampling interval.

Let $\mathbf{r}_t^{(\text{UWB})} \in \mathbb{R}^{N_t(N_t+1)/2}$ be the UWB-based ranges measured at time t that consists of robot-to-beacon ranges $[\mathbf{r}_t^{(\text{UWB})}]_i, i = 1, \dots, N_t$, and beacon-to-beacon ranges $[\mathbf{r}_t^{(\text{UWB})}]_i, i = N_t + 1, \dots, N_t(N_t + 1)/2$, and they are non-linear functions of $\mathbf{p}_t$:

$$[\mathbf{r}_t^{(\text{UWB})}]_i = h(\mathbf{p}_{j,t}, \mathbf{p}_{k,t}) = \|\mathbf{p}_{j,t} - \mathbf{p}_{k,t}\|, 0 \leq j < k \leq N_t,$$

where $i = 1, \dots, N_t(N_t + 1)/2$ index the pairwise combination of UWB nodes. We assume that all peer-to-peer ranges in $[\mathbf{r}_t]_i, i = 1, \dots, N_t(N_t + 1)/2$ are corrupted by i.i.d. additive noise $n_t \sim \mathcal{N}(0, \sigma_n^2)$, i.e., $[\hat{\mathbf{r}}_t^{(\text{UWB})}]_i = [\mathbf{r}_t^{(\text{UWB})}]_i + n_t$. This assumption has been widely adopted under line-of-sight (LOS) scenario [183].

5.3.2 EKF update

The update process proceeds via a standard EKF:

$$\begin{aligned} \hat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{w}_t, \\ \mathbf{P}_{t|t-1} &= \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_t, \\ \mathbf{S}_t &= \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \sigma_n^2 \mathbf{I}_{N_t(N_t+1)/2}, \\ \mathbf{K}_t &= \mathbf{P}_{t|t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1}, \\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(\hat{\mathbf{r}}_t^{(\text{UWB})} - h(\hat{\mathbf{x}}_{t|t-1}) \right), \\ \mathbf{P}_{t|t} &= (\mathbf{I}_{4(N_t+1)} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}, \end{aligned}$$

where $\hat{\mathbf{x}}_{t|t}$ and $\mathbf{P}_{t|t}$ are the updated state estimate and covariance estimate, respectively. The measurement matrix $\mathbf{H}_t \in \mathbb{R}^{L_t \times 4(N_t+1)}$ is defined to be the $\left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{t|t-1}}$. As

the measurements depend not on the velocity of the nodes, the partial derivatives of $h(\cdot)$ w.r.t. the velocities of robot and beacons are all zeros.

Remark 2. The estimated velocity of the robot won't be accurate if robot stops because it is derived from robot's trajectory.

5.3.3 Elimination of location ambiguities

As we are dealing with an infrastructure-less localization problem where no prior information about location of nodes is assumed, we can only derive the relative geometry of all UWB nodes based on peer-to-peer ranging measurements. Such relative geometry, as shown in Fig. 5.1, however, can be arbitrarily translated and rotated. For constructing "LiDAR map" which will be discussed in next section, we

FIGURE 5.1: Illustration of two versions of the same relative geometry of four nodes: one is translated and rotated version of the other one.

need to eliminate all ambiguous copies of the relative geometry except one. To do this, we translate and rotate the state estimate available $\hat{\mathbf{x}}_{t|t}$: $\mathbf{R}_t(\theta) \otimes \mathbf{I}_2 \begin{bmatrix} \hat{\mathbf{p}}_{t|t} \\ \hat{\mathbf{v}}_{t|t} \end{bmatrix} + \begin{bmatrix} \mathbf{T}_t \\ \mathbf{0} \end{bmatrix}$, where $\mathbf{R}_t(\theta)$ is a 2D rotation matrix and $\mathbf{T}_t \in \mathbb{R}^{2 \times (N_t+1)}$ is a translation

matrix. We find $\mathbf{R}_t(\theta)$ and $\mathbf{T}_t$ such that one beacon node is always located at origin $[0, 0]^T$ and one another beacon node is always located on positive x -axis. In addition, we force the velocity of these two chosen beacon nodes to be zero. These two beacon nodes will set up a local frame under which the LiDAR map can be built and updated.

5.3.4 Detection/removal of ranging outliers

NLOS propagation is crucial for high-resolution localization systems [184, 185] because non-negligibly positive biases will be introduced in range measurements, thus degrading the localization accuracy. A rule-of-thumb suggested by [186, 187] has been practised for NLOS detection. This rule says if the power difference, i.e. the received power minus the first-path power, is greater than 10dB the channel is likely to be NLOS. The NLOS ranges are ignored in the EKF update.

Remark 3. We have tried to detect the outliers using hypothesis testing [188] on the residuals of the EKF, i.e. $\boldsymbol{\epsilon}_t = \hat{\mathbf{r}}_t^{(\text{UWB})} - h(\hat{\mathbf{p}}_{t|t-1})$, which is a very common way to remove outliers in the infrastructure-based localization system. In absence of errors, $[\boldsymbol{\epsilon}_t]_i \sim \mathcal{N}(0, [\mathbf{S}_t]_{i,i})$. Given confidence intervals for each peer-to-peer ranging measurement, if it's violated, this range is considered an outlier and is ignored. This detection method doesn't work well in relative positioning because in the relative positioning where localization of all nodes purely relies on peer-to-peer ranges if some ranges measured at time $t - 1$ are outliers, the ranging error will be propagated to all nodes whereby the prediction ranges $\hat{\mathbf{r}}_{t|t-1}^{(\text{UWB})}$ may deviate away from the ranges $\hat{\mathbf{r}}_t^{(\text{UWB})}$ measured at time t even if $\hat{\mathbf{r}}_t^{(\text{UWB})}$ contains no outliers thus resulting in many false alarm detections.

5.4 Map Update

Due to the non-negligible accuracy gap in ranging measurement between UWB and LiDAR, constructing LiDAR map directly on top of UWB localization results is

not a promising solution. For example, if the robot keeps scanning the same object for a while, all the scan endpoints should converge to the area where the object is located. However, if the robot's pose and heading are subject to some uncertainties, say 20cm and 3 degrees, respectively, which are quite normal in UWB-only SLAM, mapping the scan endpoints according to robot's pose/heading estimates without any further process would end up with scan endpoints being diverged thereby dramatically reducing the map quality. Fig. 5.3(a) show such an example. This motivates us to align the scan endpoints by refining the robot's pose/heading estimates. Then, we turn our focus to the scan matching procedure. HectorSLAM [4] proposes a scan matching procedure based on Gauss-Newton method [189] that matches the beam endpoints observed at time t with the latest map $\mathbf{m}_{t-1}^{(\text{LiD})}$ in order to find the optimal transformation/rotation leading to the best match. The scan matching proceeds in a multi-resolution grid starting from grid maps with lower resolution to the grid maps with higher resolution so that it's more likely to find the global solution other than being trapped in local solutions. Our proposed scan matching method as will be discussed below is grounded on [4].

The pose estimates of robot and beacons $\hat{\mathbf{p}}_{t|t}$ and the velocity estimate of the robot $\hat{\mathbf{v}}_{0,t|t}$, which are obtained from EKF-based UWB-only SLAM, would be used for initializing the scan matching procedure. Note that the robot's heading information θ_t could be derived from the 2D velocity as $\hat{\theta}_t = \arctan([\hat{\mathbf{v}}_{0,t|t}]_2, [\hat{\mathbf{v}}_{0,t|t}]_1)$.

Let $\boldsymbol{\xi}_t = [(\hat{\mathbf{p}}_{t|t})^T, \hat{\theta}_t]^T$. We propose to find the optimal offset $\Delta_{\boldsymbol{\xi}_t} = [\Delta_{\mathbf{p}_t}, \Delta_{\theta_t}]^T$ such that the following objective is minimized:

$$\begin{aligned}
& \Delta_{\boldsymbol{\xi}_t}^* \\
&= \arg \min_{\Delta_{\boldsymbol{\xi}_t}} \frac{1}{2} \sum_{i=1}^n \left[1 - \Gamma(f_i \left(\hat{\mathbf{p}}_{0,t|t} + \Delta_{\mathbf{p}_{0,t}}, \hat{\theta}_t + \Delta_{\theta_t}, \hat{\mathbf{r}}_t^{(\text{LiD})} \right)) \right]^2 \\
& \quad + \frac{\gamma}{2} \sum_{0 \leq i < j \leq N_t} \left[h \left(\hat{\mathbf{p}}_{i,t|t} + \Delta_{\mathbf{p}_{i,t}}, \hat{\mathbf{p}}_{j,t|t} + \Delta_{\mathbf{p}_{j,t}} \right) \right. \\
& \quad \left. - \left[\hat{\mathbf{r}}_t^{(\text{UWB})} \right]_k \right]^2, \tag{5.1}
\end{aligned}$$

where $h(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|$ and k indexes the range measured between node i and node j and $\Delta_{\mathbf{p}_t} = \left[\Delta_{\mathbf{p}_{0,t}}^T, \dots, \Delta_{\mathbf{p}_{N_t,t}}^T \right]^T$.

Remark 4. As discussed in Section 5.3.3, two UWB beacons are used to establish the UWB coordinate system i.e. one node is always located at $[0, 0]^T$ and the other node is fixed on the positive x -axis by letting its coordinate along y -axis be zero. Hence, the values corresponding to these three coordinates in Δ_{ξ_t} are fixed to zero and we don't update them.

Now let us take a close look at (5.1).

LiDAR map matching: The first term in (5.1) intends to match the laser scan with learnt map by offsetting the robot's pose and heading. The n is the number of laser scan endpoints, and the function $f_i(\cdot)$ maps by transforming/rotating the scan endpoints, say $\mathbf{p}_i^{(\text{sc})}, i = 1, \dots, n$ which are computed based on $\hat{\mathbf{r}}_t^{(\text{LiD})}$, to their corresponding coordinates $\mathbf{p}_i^{(\text{UWB})} = f_i(\mathbf{p}_i^{(\text{sc})}), i = 1, \dots, n$ under UWB coordinate system, and the $\Gamma(\cdot)$ function returns the occupancy probabilities at coordinates $\mathbf{p}_i^{(\text{UWB})} = [x_i, y_i]^T, i = 1, \dots, n$. The occupancy probability is the probability of a grid cell where $f_i(\mathbf{p}_i^{(\text{sc})})$ is in being occupied. Follow the way proposed in [4], the occupancy probability $\Gamma(\mathbf{p}_i^{(\text{UWB})}) \in [0, 1]$ for the i th scan endpoint is approximated by bilinear interpolation using its four closest integer neighbour points in the grid map, say $\mathbf{p}_{i,j} = [x_i, y_j]^T, i, j = \{0, 1\}$. This bilinear interpolation can be written as

$$\begin{aligned} \Gamma(\mathbf{p}_i^{(\text{UWB})}) &\approx \frac{y_i - y_0}{y_1 - y_0} \left(\frac{x_i - x_0}{x_1 - x_0} \Gamma(\mathbf{p}_{1,1}) + \frac{x_1 - x_i}{x_1 - x_0} \Gamma(\mathbf{p}_{0,1}) \right) \\ &\quad + \frac{y_1 - y_i}{y_1 - y_0} \left(\frac{x_i - x_0}{x_1 - x_0} \Gamma(\mathbf{p}_{1,0}) + \frac{x_1 - x_i}{x_1 - x_0} \Gamma(\mathbf{p}_{0,0}) \right). \end{aligned}$$

The gradient of $\nabla \Gamma(f_i(\xi_t))$ is $\left[\frac{\partial \Gamma(f_i(\xi_t))}{\partial x_i}, \frac{\partial \Gamma(f_i(\xi_t))}{\partial y_i} \right]^T$. For the detailed computation of the gradient, we refer the readers to [4].

Remark 5. The UWB-only SLAM defines a relative coordinate system (Section 5.3.3 explains how it's done) on which the LiDAR map is built/updated. That's why the scan endpoints are all mapped to UWB coordinate system.

UWB map matching: The second term in (5.1) is a non-linear least square function depending on all LOS peer-to-peer UWB ranging measurements. Minimizing this function would refine the robot and beacons location estimates where γ is a tradeoff parameter. When $\gamma = 0$, (5.1) degenerates to (7) in [4] where the matching is done based on LiDAR only.

To find closed-form solution of (5.1), we approximate functions $\Gamma(\boldsymbol{\xi}_t + \Delta_{\boldsymbol{\xi}_t})$ and $h(\boldsymbol{\xi}_t + \Delta_{\boldsymbol{\xi}_t})$, respectively, by first-order Taylor expansion at point $\boldsymbol{\xi}_t$ as $\Gamma(\boldsymbol{\xi}_t + \Delta_{\boldsymbol{\xi}_t}) \approx \Gamma(\boldsymbol{\xi}_t) + \Delta_{\boldsymbol{\xi}_t}^T \nabla \Gamma(\boldsymbol{\xi}_t)$ and $h(\boldsymbol{\xi}_t + \Delta_{\boldsymbol{\xi}_t}) \approx h(\boldsymbol{\xi}_t) + \Delta_{\boldsymbol{\xi}_t}^T \nabla h(\boldsymbol{\xi}_t)$. Then, taking derivative of the objective in (5.1) w.r.t. $\Delta_{\boldsymbol{\xi}_t}$ and equating it to zero yields

$$\Delta_{\boldsymbol{\xi}_t}^* = \left(\mathbf{H}_t^{(\text{LiD})} - \gamma \mathbf{H}_t^{(\text{UWB})} \right)^{-1} \left(\mathbf{M}_t^{(\text{LiD})} + \gamma \mathbf{M}_t^{(\text{UWB})} \right), \quad (5.2)$$

where

$$\begin{aligned} \mathbf{H}_t^{(\text{LiD})} &= \sum_{i=1}^n \left[\nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t} \right] \left[\nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t} \right]^T, \\ \mathbf{H}_t^{(\text{UWB})} &= \sum_{0 \leq i < j \leq N_t} \nabla h(\hat{\mathbf{p}}_{i,t|t} + \Delta_{\mathbf{p}_{i,t}}, \hat{\mathbf{p}}_{j,t|t} + \Delta_{\mathbf{p}_{j,t}}) \\ &\quad \nabla h(\hat{\mathbf{p}}_{i,t|t} + \Delta_{\mathbf{p}_{i,t}}, \hat{\mathbf{p}}_{j,t|t} + \Delta_{\mathbf{p}_{j,t}})^T, \\ \mathbf{M}_t^{(\text{LiD})} &= \sum_{i=1}^n \left[\nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t} \right] [1 - \Gamma(f_i(\boldsymbol{\xi}_t))], \\ \mathbf{M}_t^{(\text{UWB})} &= \sum_{0 \leq i < j \leq N_t} \nabla h(\hat{\mathbf{p}}_{i,t|t} + \Delta_{\mathbf{p}_{i,t}}, \hat{\mathbf{p}}_{j,t|t} + \Delta_{\mathbf{p}_{j,t}}) \\ &\quad \left(h(\hat{\mathbf{p}}_{i,t|t} + \Delta_{\mathbf{p}_{i,t}}, \hat{\mathbf{p}}_{j,t|t} + \Delta_{\mathbf{p}_{j,t}}) - \left[\hat{\mathbf{r}}_t^{(\text{UWB})} \right]_k \right), \end{aligned}$$

where k indexes the range measured between node i and node j . Note that the optimal offset $\Delta_{\boldsymbol{\xi}_t}^*$ in (5.2) generalized the one given in (12) of [4] which considers only the LiDAR map matching.

After obtaining the optimal offset $\Delta_{\xi_t}^* = \left[(\Delta_{\mathbf{p}_t}^*)^T, \Delta_{\theta_t}^* \right]^T$, the state estimate would be corrected as

$$\begin{aligned}\hat{\mathbf{p}}_{t|t} &= \hat{\mathbf{p}}_{t|t} + \Delta_{\mathbf{p}_t}^*, \\ \hat{\mathbf{v}}_{0,t|t} &= \hat{\mathbf{v}}_{0,t|t} + \Delta_{\mathbf{v}_t}^*, \\ \Delta_{\mathbf{v}_t} &= \begin{bmatrix} \|\hat{\mathbf{v}}_{0,t|t}\| \cos \Delta_{\theta_t}^* \\ \|\hat{\mathbf{v}}_{0,t|t}\| \sin \Delta_{\theta_t}^* \end{bmatrix}.\end{aligned}$$

5.5 Experimental Results

We present here the experimental results that show the advantages of UWB/LiDAR fusion in SLAM by comparing our approach with the existing LiDAR-only SLAM approach. The robustness of UWB/LiDAR fusion in SLAM are demonstrated under different scenarios such as 1) the robot moves fast; 2) beacon(s) drop from the sensor network / new beacon(s) join the sensor network / beacons are moving; 3) the environment is feature-less. We also study the impact of γ , which tradeoffs the LiDAR map matching and UWB map matching, on the SLAM performance. All the experiments shown below share some common settings: the standard deviation (std.) of ranging/motion noise are $\sigma_n = 0.1$ meter and $\sigma_w = 0.1$ meter.

5.5.1 Hardware

Fig. 5.2(a) is our designed UWB hardware platform that integrates DWM1000 from Decawave which is used for wireless ranging and messaging. The current implementation uses 4GHz band, 850 kbps data rate at 16MHz PRF. The STM32 is controlled by the intelligence of computer/micro-computer e.g. raspberry pi, beaglebone black, etc., to execute commands given via USB. For one peer-to-peer ranging, it takes 3 milliseconds. The LiDAR sensor we use is Hokuyo UTM-30LX-EW scanning laser rangefinder. The robot i.e. a wheelchair equipped with one

FIGURE 5.2: Our UWB/LiDAR SLAM system.

UWB sensor and one LiDAR sensor is shown Fig. 6.2. The wheelchair is pushed manually in our experiments.

5.5.2 SLAM in a workshop of size $12 \times 19 \text{ m}^2$

Initially, five UWB beacons are placed at unknown locations. Note that we need to place UWB beacons accordingly to initialize the coordinate. In detail, it is better to ensure that any UWB beacon has LOS range with at least two other beacons. The robot moves at a speed about 0.8m/s along a predefined trajectory for three loops which takes about 150 seconds. Fig. 5.3 shows the UWB/LiDAR-based SLAM results. Comparing Fig. 5.3(a) with Figs. 5.3(b) to 5.3(d), we see that when there is no scan matching step the quality of the LiDAR map dramatically decreases. This explains why we cannot build LiDAR map by directly mapping the laser scan endpoints to the UWB map. Comparing Fig. 5.3(b) and Fig. 5.3(c) where the former one has no correction step (i.e. correction of the UWB map using the offset obtained in the scan matching procedure) whereas the latter one has, we can see the correction step significantly improves the estimate of robot's trajectory while the estimates of beacons' positions are roughly the same for both cases. It makes sense since the robot's pose is directly affected by LiDAR map matching,

i.e. the first term in (5.1), which however has no direct influence on beacons' poses. Fig. 5.3(d) ignores the UWB map matching, i.e. the second term in (5.1), by setting γ to be a negligibly small value 10^{-6} . Comparing Fig. 5.3(d) with Fig. 5.3(c), we can see the localization for both robot and beacons are distorted when the role of second term in (5.1) is downplayed. Table 5.2 gives the averaged (over time steps

FIGURE 5.3: UWB/LiDAR fusion in four cases: a) no scan matching, no correction; b) with UWB/LiDAR scan matching where $\gamma = 0.65$, no correction; c) with UWB/LiDAR scan matching where $\gamma = 0.65$ and correction; d) with LiDAR-only scan matching where $\gamma = 10^{-6}$ and correction. The green "+" denotes the final position of the robot.

and over five nodes) positioning error and std. of UWB beacons. The estimates of five beacons' poses are recorded while robot is moving. From the table we can see choosing $\gamma = 0.65$ and enabling the correction step gives best results among the three settings. In the following experiments, we empirically choose $\gamma = 0.65$ but we believe the choice of γ is environmentally dependent. A simple rule-of-thumb is that let γ be large when the environments have less features and/or there are

sufficient LOS UWB ranges, and let γ be small when the environments are feature-sufficient and/or there are only few LOS UWB ranges. Note that the difference between "no correction" and "w/ correction" is whether the "Correction" module in Fig. 1.4 works or not when building the map for the environment.

TABLE 5.2: Averaged errors/stds. of five UWB beacons' pose estimates.

	$\gamma = 0.65$ no correction	$\gamma = 0.65$ w/ correction	$\gamma = 10^{-6}$ w/ correction
err. in meters	0.213	0.076	0.206
std. in meters	0.136	0.122	0.149

Fig. 5.4 demonstrates our system is capable of handling dynamical scenarios where the beacons may drop/join/move. We divide the SLAM process into three time slots: a) For $t < 135$ (135 time steps not 135 seconds), there are four static beacons; b) We power on a new beacon (with ID. 5) at $t = 135$ at the "start pos." shown in Fig. 5.4(a), then this new beacon is moved along the trajectory of the robot until we place it to the "end pos." at $t = 311$; c) We power off one existing beacon (with ID. 3) at $t = 311$ and relocate it to the "start pos." shown in Fig. 5.4(b), then power it on at $t = 466$ and move it along the trajectory of the robot until we place it to "end pos." at $t = 626$. Fig. 5.4(c) shows how the number beacons varies over time steps. In Fig. 5.5, we compare UWB/LiDAR-based SLAM with HectorSLAM (LiDAR-only SLAM) when robot moves at different speeds. When the robot moves at about 0.4m/s, HectorSLAM is working properly and a high-quality map shown in Fig. 5.5(a) is constructed. However, when the robot moves faster at about 0.8m/s, HectorSLAM becomes error-prone where the errors mainly come from fast turnings as HectorSLAM cannot accommodate a sudden change of robot's behaviour especially the change of heading, and these errors would affect the subsequent scan matching steps thus being accumulated over time. This can be seen in Fig. 5.5(b). Such errors can be dramatically reduced due to additional inputs from UWB sensors. In Fig. 5.5(c) where robot moves as fast as the one in Fig. 5.5(b), we observe no drift-errors in UWB/LiDAR-based SLAM.

(a) $135 \leq t \leq 311$ (b) $466 \leq t \leq 626$ (c) N_t

FIGURE 5.4: Beacon(s) drops/joins/moves while SLAM is proceeding.

(a) HectorSLAM at 0.4m/s

(b) HectorSLAM at 0.8m/s

(c) UWB/LiDAR at 0.8m/s

FIGURE 5.5: Comparison of our UWB/LiDAR-based SLAM with HectorSLAM [4] at different robot's speeds. To build the maps, the robot moves for one loop of the same trajectory as shown in Fig. 5.3.

5.5.3 SLAM in a corridor of length 22.7 meters

HectorSLAM is known to work poorly in feature-less environments such as corridor and Fig. 5.6(a) shows such an example where the length of corridor in its map is 8.43m whereas the actual length is 22.7m. Fusing LiDAR with UWB, we may regard UWB sensors as additional “features” that provide precise information about robot’s location thus robustifying SLAM in such feature-less environments at the cost of slight degradation of map quality as shown in Fig. 5.6(b). Note that there still exist errors in the measured length of corridor (22.58 meters) compared to the ground truth (22.7 meters). It is because that the map is built upon the coordinate of UWB beacons, so the error in UWB beacons will be transferred to the built map, even with the proposed correction on UWB beacons.

(a) HectorSLAM

(b) UWB/LiDAR SLAM

FIGURE 5.6: Comparison of our UWB/LiDAR-based SLAM with HectorSLAM [4] in a corridor of length 22.7m.

Remark 6. The quality of the map that UWB/LiDAR-based SLAM builds is compromised due to the tradeoff between UWB and LiDAR ranging accuracy which is reflected from (5.1). Moreover, to balance the UWB and LiDAR ranging accuracy, we add some Gaussian noise with zero mean and std. of 1cm/2cm to the laser ranger finder.

5.6 Conclusion

We have proposed a fusion scheme that utilizes both UWB and laser ranging measurements for simultaneously localizing the robot, the UWB beacons and constructing the LiDAR map. This is a 2D range-only SLAM and no control input is required. In our fusion scheme, a "coarse" map i.e. UWB map is built using UWB peer-to-peer ranging measurements. This "coarse" map can guide where the "fine" map i.e. LiDAR map should be built. Through a new scan matching procedure, the LiDAR map can be constructed while the UWB map is polished. Our proposed system suits infrastructure-less and ad-hoc applications where no prior information about the environment is known and the system needs to be deployed quickly.

Chapter 6

UWB/LiDAR Fusion SLAM via Step-by-step Iterative Optimization

6.1 Introduction

The fusion SLAM scheme presented in Chapter 5 has two limitations. Firstly, the closed-form solution is achieved by approximately linearizing the non-linear objective function. Such approximation requires the states variation of robot and UWB beacons be small (i.e. a slow motion of the robot). For a fast moving robot, the error is prone to being accumulated. Secondly, a composite loss, i.e. sum of UWB loss and LiDAR loss, is formulated to optimize the states of robot and UWB beacons simultaneously. Thus, the trade-off between these two sensors, which have different accuracy, needs to be manually tuned. Meanwhile, we note that such composite optimization is not so effective to improve the UWB beacons' state estimations as the LiDAR measurements have no direct impact on the UWB beacons. Hence, we present an improved fusion scheme in this chapter which refines the state of robot and UWB beacons step-by-step, i.e. firstly refine robot's states

using LiDAR range measurements and then rectifies the UWB beacons' states. The key features of the proposed system are highlighted below:

1. It introduces the idea of particle filter and trust-region optimization to fuse UWB/LiDAR measurements.
2. It involves a new scan matching procedure to iteratively tune the robot's states that are initially estimated using UWB ranges to match with LiDAR measurements.
3. It has a beacon correction step after every scan matching procedure so that the refined information can be propagated from robot to beacons.

6.2 Problem Formulation

There are three unknown quantities we want to optimize with: 1) $\mathbf{M}_{\rightarrow t}$, the LiDAR map learnt up to time t , 2) $\mathbf{p}_{1:t}^i \in \mathbb{R}^2, i = 1, \dots, N$, the i -th UWB node's 2D location up to time t and 3) $\mathbf{x}_{1:t,r} \in \mathbb{R}^3$, the robot's state up to time t containing its 2D location and heading direction.

We consider three sets of observations: 1) $\mathbf{z}_{1:t}^{\text{UWB}}$ including pairwise ranges between UWB nodes measured up to time t , 2) $\mathbf{u}_{1:t-1}^r$ including control vectors measured up to time $t - 1$ by encoders and 3) $\mathbf{p}_{1:t}^{\text{sc}}$ including LiDAR scan end-points up to time t .

The objective of the UWB/LiDAR based SLAM is to estimate the following posterior

$$p(\{\mathbf{p}_{1:t}^i\}_{i=1}^N, \mathbf{x}_{1:t,r}, \mathbf{M}_{\rightarrow t} | \mathbf{z}_{1:t}^{\text{UWB}}, \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{u}_{1:t-1}^r). \quad (6.1)$$

Following the idea of RBPF, (6.1) can be factorized into

$$p(\{\mathbf{p}_{1:t}^i\}_{i=1}^N, \mathbf{x}_{1:t,r} | \mathbf{z}_{1:t}^{\text{UWB}}, \mathbf{u}_{1:t-1}^r) \cdot p(\mathbf{M}_{\rightarrow t} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r}), \quad (6.2)$$

where the first term defines a localization problem with the objective of localizing all N UWB nodes as well as the robot based on UWB ranging measurements only, while the second term can be computed efficiently since the robot state $\mathbf{x}_{1:t,r}$ are known when estimating the map $\mathbf{M}_{\rightarrow t}$. Fig. 1.5 presents the overall structure of the proposed autonomous exploration system. In what follows, Section 6.3 deals with the estimation of the first term in (6.2) while Section 6.4 tackles the estimation of the second term.

6.3 Localization Using UWB Measurements Only

The objective of this section is to estimate the first term in (6.2), i.e., $p(\{\mathbf{p}_{1:t}^i\}_{i=1}^N, \mathbf{x}_{1:t,r} | \mathbf{z}_{1:t}^{\text{UWB}}, \mathbf{u}_{1:t-1}^r)$, which can be factorized into

$$p(\{\mathbf{p}_{1:t}^i\}_{i=1}^N | \mathbf{z}_{1:t}^{\text{UWB}}) \cdot p(\mathbf{x}_{1:t,r} | \{\mathbf{p}_{1:t}^i\}_{i=1}^N, \mathbf{u}_{1:t-1}^r). \quad (6.3)$$

6.3.1 EKF-based Range-Only Localization

The estimation of UWB nodes' states, i.e., $p(\{\mathbf{p}_{1:t}^i\}_{i=1}^N | \mathbf{z}_{1:t}^{\text{UWB}})$, is performed using a standard EKF which incrementally processes the observations. The EKF proceeds as follows

$$\mathbf{m}_{t|t-1} = \mathbf{F}_t \mathbf{m}_{t-1|t-1} + \boldsymbol{\nu}_t, \quad (6.4)$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_t, \quad (6.5)$$

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t, \quad (6.6)$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1}, \quad (6.7)$$

$$\mathbf{m}_{t|t} = \mathbf{m}_{t|t-1} + \mathbf{K}_t (\mathbf{z}_t^{\text{UWB}} - h(\mathbf{m}_{t|t-1})), \quad (6.8)$$

$$\mathbf{P}_{t|t} = (\mathbf{I}_{4N} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1}, \quad (6.9)$$

where $\mathbf{m}_{t|t}$ denotes the updated state estimation that includes the position $\{\mathbf{p}_t^i\}_{i=1}^N$ and velocity $\{\mathbf{v}_t^i\}_{i=1}^N$ of UWB nodes, $\mathbf{P}_{t|t}$ denotes the updated covariance estimate,

$\mathbf{F}_t$ denotes the transition matrix which is equal to $\begin{bmatrix} \mathbf{I}_{2N} & \tau \mathbf{I}_{2N} \\ 0 & \mathbf{I}_{2N} \end{bmatrix}$, $\boldsymbol{\nu}_t$ is the state noise with zero-mean and covariance $\mathbf{Q}_t = \begin{bmatrix} \tau^2 \mathbf{I}_{2N} & \tau \mathbf{I}_{2N} \\ \tau \mathbf{I}_{2N} & \mathbf{I}_{2N} \end{bmatrix}$ and τ is the sampling interval [190]. The measurement matrix $\mathbf{H}_t \in \mathbb{R}^{L \times 4N}$ is defined to be $\frac{\partial h}{\partial \mathbf{m}} \big|_{\mathbf{m}_{t|t-1}}$, where $L = N(N-1)/2$ is the number of pairwise ranges and $\mathbf{R}_t$ is the covariance matrix regarding the UWB range measurements.

6.3.2 PF-Based Robot's State Estimation

Denote $\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2$ as the estimated $\{\mathbf{p}_{1:t}^i\}_{i=1}^2$ obtained from EKF. The robot's state $\mathbf{x}_{t,r}$ relates to $\mathbf{p}_t^1$ and $\mathbf{p}_t^2$ are as follows

$$\mathbf{x}_{t,r} = g(\mathbf{p}_t^1, \mathbf{p}_t^2) = \begin{bmatrix} (\mathbf{p}_t^1 + \mathbf{p}_t^2)/2 \\ \arctan(\mathbf{p}_t^2 - \mathbf{p}_t^1) + \pi/2 \end{bmatrix}, \quad \forall t \quad (6.10)$$

This relation is due to the fact that the first two UWB nodes are mounted symmetrically on the robot. Please refer to Fig. 6.2(a) for the physical set-up. We can directly compute $\mathbf{x}_{1:t,r}$ using (6.10). However, since the separation between the two on-board UWB nodes is about the width of the robot which is short, a noisy estimation of $\{\mathbf{p}_{1:t}^i\}_{i=1}^2$ would yield large variance in estimating the robot's state $\mathbf{x}_{1:t,r}$. This motivates us to make the following assumption.

Assumption 1. The mapping from $\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2$ to $\mathbf{x}_{t,r}$ is corrupted by an additive Gaussian noise with zero mean and covariance matrix $\boldsymbol{\Sigma}_x \in \mathbb{R}^{3 \times 3}$. That is,

$$\mathbf{x}_{t,r} = g(\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2) + \mathbf{n}_{t,r}, \quad \forall t, \quad (6.11)$$

with $\mathbf{n}_{t,r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_x)$.

Based on Assumption 1, we have, for all t ,

$$p(\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2 | \mathbf{x}_{t,r}) \propto \exp^{-\frac{1}{2}(\mathbf{x}_{t,r} - g(\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2))^T \boldsymbol{\Sigma}_x^{-1} (\mathbf{x}_{t,r} - g(\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2))}. \quad (6.12)$$

We propose to estimate the second term in (6.3), i.e., $p(\mathbf{x}_{1:t,r}|\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2, \mathbf{u}_{1:t-1}^r)$, using a common sampling importance resampling (SIR) filter. Following [191], we can make the following approximation

$$p(\mathbf{x}_{1:t,r}|\{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2, \mathbf{u}_{1:t-1}^r) \approx \sum_{j=1}^{N_{p_1}} w_t^{(j)} \delta(\mathbf{x}_{1:t,r} - \mathbf{x}_{1:t,r}^{(j)}), \quad (6.13)$$

where $\delta(x)$ denotes Dirac delta function, i.e., $\delta(x) = 1$ if $x = 0$ or 0 otherwise, and

$$w_t^{(j)} \propto \frac{p(\{\hat{\mathbf{p}}_t^i\}_{i=1}^2|\mathbf{x}_{t,r}^{(j)}) \cdot p(\mathbf{x}_{t,r}^{(j)}|\mathbf{x}_{t-1,r}^{(j)}, \mathbf{u}_{t-1}^r)}{\pi(\mathbf{x}_{t,r}^{(j)}|\mathbf{x}_{t-1,r}^{(j)}, \{\hat{\mathbf{p}}_{1:t}^i\}_{i=1}^2, \mathbf{u}_{1:t-1}^r)} \cdot w_{t-1}^{(j)}, \quad (6.14)$$

with $p(\{\hat{\mathbf{p}}_t^i\}_{i=1}^2|\mathbf{x}_{t,r}^{(j)})$ being the likelihood of $\{\hat{\mathbf{p}}_t^i\}_{i=1}^2$ given $\mathbf{x}_{t,r}^{(j)}$, and $p(\mathbf{x}_{t,r}^{(j)}|\mathbf{x}_{t-1,r}^{(j)}, \mathbf{u}_{t-1}^r)$ being the prior that considers the robot's dynamics. It is often convenient to choose the importance density to be the prior, that is, $\pi(\mathbf{x}_{t,r}^{(j)}|\mathbf{x}_{t-1,r}^{(j)}, \{\hat{\mathbf{p}}_t^i\}_{i=1}^2, \mathbf{u}_{1:t-1}^r) = p(\mathbf{x}_{t,r}^{(j)}|\mathbf{x}_{t-1,r}^{(j)}, \mathbf{u}_{t-1}^r)$, where the control vector $\mathbf{u}_t^r = [v_t, \omega_t]^T \in \mathbb{R}^2$ includes the translational velocity v_t and rotational velocity ω_t . A velocity motion model is applied to estimate the next generation's particles. Please refer to Section 5.3 in [192] for more details.

Then, (6.14) becomes

$$w_t^{(j)} = w_{t-1}^{(j)} \cdot p(\{\hat{\mathbf{p}}_t^i\}_{i=1}^2|\mathbf{x}_{t,r}^{(j)}). \quad (6.15)$$

The implementation of this particle filter is summarized in the following three steps:

1. **Sampling:** A proposal distribution π is defined to sample the next generation of particles $\{\mathbf{x}_{t,r}^{(j)}\}_{j=1}^{N_{p_1}}$ based on the current generation $\{\mathbf{x}_{t-1,r}^{(j)}\}_{j=1}^{N_{p_1}}$, where N_{p_1} is the number of robot's coarse particles.
2. **Weighing of particles:** Each particle is assigned an importance weight $w_t^{(j)}$ according to the importance evaluation metric.

3. **Resampling:** To handle unavoidable degeneracy of the importance sampling, particles will be redrawn when necessary based on the particles' weights. Therefore, the particles with small normalised weight will be eliminated.

At each time step t , the robot's state estimation is a weighted summation of all the particles

$$\hat{\mathbf{x}}_{t,r} = \frac{w_t^{(j)} \cdot \mathbf{x}_{t,r}^{(j)}}{\sum_{j=1}^{N_{p1}} w_t^{(j)}}, \quad (6.16)$$

which avoids abrupt change of the robot's state estimation, especially the bearing estimation, over two time steps.

6.4 Scan Matching and Mapping

This section is to estimate the second term of (6.2), i.e., $p(\mathbf{M}_{\rightarrow t} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r})$, which can be approximated by

$$p(\mathbf{M}_{\rightarrow t} | \mathbf{p}_{1:t}^{\text{sc}}, \hat{\mathbf{x}}_{1:t,r}), \quad (6.17)$$

where $\hat{\mathbf{x}}_{1:t,r}$ are estimated in (6.16).

It is worth noting that $\hat{\mathbf{x}}_{1:t,r}$ are estimated using UWB ranging measurements only while the mapping process mainly uses the LiDAR measurements. However, there exists a precision gap between UWB ranging and LiDAR ranging. One can minimize this gap by exhaustively searching for the robot's optimal state, which yields the best match-up between $\mathbf{p}_t^{\text{sc}}$ and the $\mathbf{M}_{\rightarrow t-1}$, around $\hat{\mathbf{x}}_{t,r}$. Such searching strategy is computationally forbidden. Therefore, we propose an adaptive scan matcher to efficiently fine-tune the robot's state while the searching region is adaptively adjusted.

Before we introduce the proposed scan matching scheme, here are some preliminaries. Similar to [193], the alignment loss of scan matching is defined as:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2n} \sum_{i=1}^n [1 - \Gamma(\mathbf{p}_{t,i}^w)]^2, \quad (6.18)$$

where $\mathbf{p}_{t,i}^w, i = 1, \dots, n$, is the i -th scan end-point's position in the UWB frame which defines the world coordinate symbolized by w and n is the number of scan end-points. We compute $\mathbf{p}_{t,i}^w$ as $\mathbf{p}_{t,i}^w = f(\mathbf{x}_{t,r}, \mathbf{p}_{t,i}^{\text{sc}})$, where $\mathbf{p}_{t,i}^{\text{sc}}$ denotes the position of the i -th scan end-point in robot's coordinate and $f(\mathbf{x}_{t,r}, \mathbf{p}_{t,i}^{\text{sc}})$ rotates and translates $\mathbf{p}_{t,i}^{\text{sc}}$ to the world coordinate. The function $\Gamma(\mathbf{p}_{t,i}^w)$ interpolates the occupancy probability at $\mathbf{p}_{t,i}^w$ using the method proposed in [4]. Therefore, a smaller $\mathcal{L}$ corresponds to a better alignment between scan end-points and the built map.

The objective of scan matching is to find an optimal offset $\Delta_{\mathbf{x}}^* = \arg \min_{\Delta_{\mathbf{x}}} \mathcal{L}(\mathbf{x})$. Since (6.18) is non-linear, an analytic solution of $\Delta_{\mathbf{x}}$ is obtained in [193] through approximating $f(\mathbf{x} + \Delta_{\mathbf{x}}, \mathbf{p}_{t,i}^{\text{sc}})$ by its first-order Taylor expansion evaluated at $\mathbf{x}$. This approximation requires slow motion of the robot, as the actual $\Delta_{\mathbf{x}}$ should be small. For a fast moving robot, error accumulation occurs. Therefore, we propose a PF-based recursive scan matching procedure which avoids linearizing $\mathcal{L}$ in (6.18).

6.4.1 Fine-tuning the Robot's State Using Scan Matching

The proposed recursive scan matching at time t is to maximize the likelihood of the robot's state at iteration k relative to the robot's state at iteration $k - 1$

$$\mathbf{x}_k^* = \arg \max_{\mathbf{x}_k} p(\mathbf{p}_t^{\text{sc}} | \mathbf{x}_k, \mathbf{M}_{\rightarrow t-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}^*), \quad (6.19)$$

where $0 < k \leq I_{\text{stop}}$ and $\mathbf{x}_0^* = \hat{\mathbf{x}}_{t,r}$.

To compute the prior $p(\mathbf{x}_k | \mathbf{x}_{k-1}^*)$, we make the following assumption.

Assumption 2. The $\mathbf{x}_k$ follows a Gaussian distribution with mean $\mathbf{x}_{k-1}^* \in \mathbb{R}^{3 \times 1}$ and covariance matrix $\Sigma_k \in \mathbb{R}^{3 \times 3}$.

FIGURE 6.1: An illustration of the proposed adaptive-trust-region scan matcher. The optimization starts at the robot's coarse state estimation and then a group of particles are sampled based on an adaptively learned proposal distribution. At each iteration, the optimal particle (i.e., the one with the largest weight) is found and set as the initial position for the next iteration. The red circle denotes the approximate size of search region.

The (2) restricts the searching region at iteration k to be the neighbourhood of $\mathbf{x}_{k-1}^*$. If Σ_k is sufficiently large, the local optimum of (6.19) can be escaped with high probability. We will return to how to choose Σ_k later.

The objective function in (6.19) can be approximated as

$$\sum_j^{N_{p_2}} \vartheta_k^{(j)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(j)}). \quad (6.20)$$

The weights $\{\vartheta_k^{(j)}\}_{j=1}^{N_{p_2}}$ are chosen using the principle of importance sampling [194] as

$$\vartheta_k^{(j)} \propto p(\mathbf{p}_t^{\text{sc}} | \mathbf{x}_k^{(j)}, \mathbf{M}_{\rightarrow t-1}) \cdot \vartheta_{k-1}^{(j)}, \quad (6.21)$$

where the particles are drawn according to a proposal distribution π_k^{ref} which we choose to be $p(\mathbf{x}_k | \mathbf{x}_{k-1}^*)$. Due to (2), $\pi_k^{\text{ref}} = \mathcal{N}(\mathbf{x}_{k-1}^*, \Sigma_k)$. Since the particles $\{\mathbf{x}_k^{(j)}\}_{j=1}^{N_{p_2}}$ at the k -th iteration are drawn according to $\mathcal{N}(\mathbf{x}_{k-1}^*, \Sigma_k)$, the weights computed from the previous iteration do not affect the weights' computation at

the current iteration and thus (6.21) becomes

$$\vartheta_k^{(j)} \propto p(\mathbf{p}_t^{\text{sc}} | \mathbf{x}_k^{(j)}, \mathbf{M}_{\rightarrow t-1}), \quad (6.22)$$

and (6.19) is equivalent to

$$\mathbf{x}_k^* = \arg \max_{\mathbf{x}_k^{(j)}} \vartheta_k^{(j)}. \quad (6.23)$$

Now the only remaining problem is to evaluate $p(\mathbf{p}_t^{\text{sc}} | \mathbf{x}_k^{(j)}, \mathbf{M}_{\rightarrow t-1})$. We propose to evaluate it based on the loss function defined in (6.18):

$$p(\mathbf{p}_t^{\text{sc}} | \mathbf{x}_k^{(j)}, \mathbf{M}_{\rightarrow t-1}) \propto \exp^{-\mathcal{L}(\mathbf{x}_k^{(j)})}. \quad (6.24)$$

The iteration continues until the stop criteria is met, i.e., $\mathcal{L}(\mathbf{x}_k) \leq \mathcal{L}_{\text{stop}}$, or the maximum number of iterations is reached, i.e., $k = I_{\text{stop}}$. This process is illustrated in (6.1). The optimal $\mathbf{x}_{t,r}$ would be $\mathbf{x}_{t,r} = \mathbf{x}_k^*$, where k is such that $\mathcal{L}(\mathbf{x}_k^*) \leq \mathcal{L}_{\text{stop}}$. In case there exists no such $\mathbf{x}_k^*$, we do not proceed to the mapping process.

One key ingredient in above mentioned scan matching scheme is the strategy to determine the covariance matrix of the proposal distribution π_k^{ref} , i.e., Σ_k . Motivated by the search strategy applied by trust region optimization [195], it is desired to adaptively change the searching region according to the evaluation on the optimization process. We propose to determine Σ_k by letting the variances be proportional to the difference between $\mathcal{L}(\mathbf{x}_{k-1}^*)$ and the desired loss $\mathcal{L}_{\text{stop}}$ and inversely proportional to $\mathcal{L}(\mathbf{x}_{k-2}^*) - \mathcal{L}(\mathbf{x}_{k-1}^*)$. Based on this intuition, we define a hyper parameter ρ_k as

$$\rho_k = \frac{\mathcal{L}(\mathbf{x}_{k-1}^*) - \mathcal{L}_{\text{stop}}}{\mathcal{L}(\mathbf{x}_{k-2}^*) - \mathcal{L}(\mathbf{x}_{k-1}^*) + \varsigma}, \quad (6.25)$$

where ς is a small positive value, e.g., 10^{-4} , to prevent numerical instability when the denominator is close to zero. Note that $\mathcal{L}(\mathbf{x}_{k-2}^*) \geq \mathcal{L}(\mathbf{x}_{k-1}^*)$ always holds because the best particle found in the current iteration is retained in the next

iteration, and $\mathcal{L}(\mathbf{x}_{k-1}^*) \geq \mathcal{L}_{\text{stop}}$ before iteration ends. Since $\mathcal{L}(\mathbf{x}_{k-1}^*) \leq \mathcal{L}_{\text{stop}}$ means that the optimal scan matching is found, thus $\rho_k \geq 0$. The numerator of (6.25) indicates the distance to the desired loss, and the denominator is the loss reduction between two consecutive iterations. Therefore, $\rho_k \rightarrow 0$ means that the algorithm almost reaches the optimal state, then we need to downsize the searching region. If $\rho_k \rightarrow 1$, it is safe to keep current searching region for the next iteration. If $\rho_k \gg 1$, it means that the loss reduction is too small while it is still far away from the desired loss $\mathcal{L}_{\text{stop}}$ so we need to upsize the searching region. We then use ρ_k to construct $\Sigma_k = \Sigma_{\text{base}} \cdot \rho_k$, where $\Sigma_1 = \Sigma_{\text{base}}$ is the base covariance matrix.

6.4.2 Mapping

After fine-tuning $\hat{\mathbf{x}}_{1:t,r}$, (6.17) becomes

$$p(\mathbf{M}_{\rightarrow t} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r}). \quad (6.26)$$

The map is represented by an evenly distributed occupancy grid, where each grid cell is a binary random variable that models the occupancy. Assume the mappings of each cell are independent of each other, (6.26) can be written as $\prod_{x,y} p([\mathbf{M}_{\rightarrow t}]_{x,y} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r})$, where $[\mathbf{M}_{\rightarrow t}]_{x,y}$ denotes the cell at (x, y) in $\mathbf{M}_{\rightarrow t}$. Following [196], the mapping process can be recursively computed as

$$\begin{aligned} & \log \left(\frac{p([\mathbf{M}_{\rightarrow t}]_{x,y} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r})}{1 - p([\mathbf{M}_{\rightarrow t}]_{x,y} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r})} \right) \\ &= \log \left(\frac{p([\mathbf{M}_{\rightarrow t}]_{x,y} | \mathbf{p}_t^{\text{sc}}, \mathbf{x}_{t,r})}{1 - p([\mathbf{M}_{\rightarrow t}]_{x,y} | \mathbf{p}_t^{\text{sc}}, \mathbf{x}_{t,r})} \right) \\ & \quad + \log \left(\frac{p([\mathbf{M}_{\rightarrow t-1}]_{x,y} | \mathbf{p}_{1:t-1}^{\text{sc}}, \mathbf{x}_{1:t-1,r})}{1 - p([\mathbf{M}_{\rightarrow t-1}]_{x,y} | \mathbf{p}_{1:t-1}^{\text{sc}}, \mathbf{x}_{1:t-1,r})} \right), \end{aligned} \quad (6.27)$$

where $p([\mathbf{M}_{\rightarrow t}]_{x,y} | \mathbf{p}_t^{\text{sc}}, \mathbf{x}_{t,r})$ is a correction probability due to the current measurement. The update is performed in log domain for high computational efficiency since it only requires to compute sums. We only update the cells which are in the

perceptual field of $\mathbf{p}_t^{\text{sc}}$ while the other cells remain unchanged. In our experiments, we set it to be 0.7. The process of proposed RASM is given in Algorithm 2.

Algorithm 2: The proposed adaptive trust region scan matcher

Require: $\mathbf{x}_0^* = \hat{\mathbf{x}}_{t,r}, \mathbf{p}_t^{\text{sc}}, \mathbf{M}_{\rightarrow t-1}$

- 1: $k \leftarrow 1, \Sigma_1 = \Sigma_{\text{base}}$
- 2: **while** $\mathcal{L}(\mathbf{x}_{k-1}^*) \geq \mathcal{L}_{\text{stop}}$ **and** $k \leq I_{\text{stop}}$ **do**
- 3: **if** $k \geq 2$ **then**
- 4: $\rho_k \leftarrow \frac{\mathcal{L}(\mathbf{x}_{k-1}^*) - \mathcal{L}_{\text{stop}}}{\mathcal{L}(\mathbf{x}_{k-2}^*) - \mathcal{L}(\mathbf{x}_{k-1}^*) + \varsigma}$
- 5: $\Sigma_k \leftarrow \Sigma_{\text{base}} \cdot \rho_k,$
- 6: **end if**
- 7: $\{\mathbf{x}_k^{(j)}\}_{j=1}^{N_{p2}} \sim \mathcal{N}(\mathbf{x}_{k-1}^*, \Sigma_k)$
- 8: $\vartheta_k^{(j)} \leftarrow \exp^{-\frac{1}{2n} \sum_{i=1}^n [1 - \Gamma(f(\mathbf{x}_k^{(j)}, \mathbf{p}_t^{\text{sc}}))]}]^2$
- 9: $\mathbf{x}_k^* \leftarrow \arg \max_{\mathbf{x}_k^{(j)}} \vartheta_k^{(j)}$
- 10: $k \leftarrow k + 1$
- 11: **end while**
- 12: $\mathbf{x}_{t,r} \leftarrow \mathbf{x}_{k-1}^*$
- 13: **if** $\mathcal{L}(\mathbf{x}_{k-1}^*) \leq \mathcal{L}_{\text{stop}}$ **then**
- 14: Update $p(\mathbf{M}_{\rightarrow t} | \mathbf{p}_{1:t}^{\text{sc}}, \mathbf{x}_{1:t,r})$ by (6.27)
- 15: **end if**

6.4.3 State Correction

With the refined robot's state $\mathbf{x}_{t,r}$, the states of on-board nodes and beacons can be corrected. As illustrated in Fig. 1.5, we first correct the on-board UWB nodes' states $\{\mathbf{p}_t^i\}_{i=1}^2$ by

$$\mathbf{p}_t^1 = \mathbf{x}_{t,r}(1:2) + \begin{bmatrix} -\frac{D}{2} \sin(\mathbf{x}_{t,r}(3)) \\ \frac{D}{2} \cos(\mathbf{x}_{t,r}(3)) \end{bmatrix}, \quad (6.28)$$

$$\mathbf{p}_t^2 = \mathbf{x}_{t,r}(1:2) + \begin{bmatrix} \frac{D}{2} \sin(\mathbf{x}_{t,r}(3)) \\ -\frac{D}{2} \cos(\mathbf{x}_{t,r}(3)) \end{bmatrix}, \quad (6.29)$$

where D is the separation between two on-board UWB nodes. Then, we correct UWB beacons' states based on the corrected on-board nodes' states $\{\mathbf{p}_t^i\}_{i=1}^2$ and the UWB range measurement $\mathbf{z}_t^{\text{UWB}}$ by another EKF. Note that $\mathbf{z}_t^{\text{UWB}}$ includes two parts: beacon-to-beacon ranging $\mathbf{z}_t^{\text{b-b}}$ and beacon-to-robot (i.e., the nodes on

the robot) ranging $\mathbf{z}_t^{\text{b-r}}$. We do not use $\mathbf{z}_t^{\text{b-b}}$ for beacon correction so we ignore them by setting their corresponding variances to be sufficiently high. Finally, the importance weight $w_t^{(j)}$ of each particle is updated with the corrected robot's state $\mathbf{x}_{t,r}$. The whole process of the proposed fusion scheme is given in Fig. 1.5.

6.5 Experimental Results

We evaluate our system in two environments: a cluttered indoor workshop and a spacious outdoor garden. Our experiments are conducted from the following five aspects: 1) the proposed fusion SLAM vs. existing fusion SLAM; 2) the accuracy of estimating UWB beacons' states with the proposed method vs. the fusion introduced in previous section; 3) evaluating the effects of different separations between two on-board UWB nodes on the SLAM.

6.5.1 Experimental Environment and Parameters Selection

Fig. 6.2(a) shows a robot of dual-UWB electronic wheelchair that integrates 1) two Decawave UWB sensors which are used for wireless ranging and messaging; 2) a Hokuyo UTM-30LX-EW 2D LiDAR which is used to sense the surrounding obstacles; 3) two encoders which provide the motion state of the robot, 4) an Arduino Mega board which is used to control the wheelchair; 5) a laptop which runs the core algorithm. Two experimental scenarios (workshop and garden) are shown in Figs. 6.2(b) to 6.2(d). Five and six UWB beacons are placed at unknown locations in workshop and garden scenarios, respectively. Two experimental scenarios (workshop and garden) are shown in Figs. 6.2(b) to 6.2(d). Five and six Decawave UWB beacons are placed at unknown locations in workshop of size $12 \times 19\text{m}^2$ and garden scenarios of size $45 \times 30\text{m}^2$, respectively.

FIGURE 6.2: Hardware platform and experimental physical environment.

Table 6.1 provides a guideline for selecting the parameters. Note that, for the colour representations shown in the figures in Section 7.6, grey, black and white represent unexplored, free and occupied region, respectively.

Remark 7. A calibration phase is taken to initialize the UWB beacons' states while all beacons are static. During this phase, we collect each UWB beacon's location estimate (using EKF) from the latest T time steps to compute the location variance for each beacon. A beacon is considered to be calibrated if its location variance is less than a given threshold σ_e^2 , i.e. 0.25. Once the beacon is calibrated, we start calibrating the next beacon.

The parameters are set as the same for all the following experiments. The parameters for EKF follow from Chapter 5. The parameters used in Algorithm 2, like δ_{base} , $\mathcal{L}_{\text{stop}}$, Σ_{min} , Σ_{max} , I_{stop} and ς are set to be $\text{diag}(0.04, 0.04, 0.03)$ where

TABLE 6.1: Guideline for the parameter tuning, where $\uparrow$ indicates increasing in value while $\downarrow$ means decreasing in value.

	Functions	Effects	Value
Robot's coarse state estimation (Section 6.3)			
Σ_x	observation noise of the robot's state achieved from EKF	$\Sigma_x \uparrow$: trusts more on robot's motion model, $\Sigma_x \downarrow$: trusts more on EKF estimation	$diag(0.45^2, 0.45^2, 0.17^2)$ (depending on specific UWB)
RASM (Section 6.4)			
Σ_{base}	base distribution which is set based on the accuracy gap between UWB and LiDAR range	$\Sigma_{base} \uparrow$: increase the searching region or vice versa	$diag(0.2^2, 0.2^2, 0.17^2)$
$\mathcal{L}_{stop}$	the maximum alignment loss of a successful scan matching	$\mathcal{L}_{stop} \uparrow$: lower matching accuracy but faster RASM process or vice versa	0.1 (about 90% of laser points are well aligned with map)
I_{stop}	maximum number of iterations	$I_{stop} \uparrow$: higher chance to find the optimal solution but slower RASM process or vice versa	12
$N_{p1,2}$	number of particles in Section 6.3.2 and Section 6.4.1	$N_{p1,2} \uparrow$: more accurate state estimation but slower PF computation or vice versa	200

$diag(\cdot)$ returns a square diagonal matrix with the given entries on the main diagonal, 0.1, $0.01\mathbf{I}_{3 \times 3}$, $0.06\mathbf{I}_{3 \times 3}$, 12 and $1e^{-4}$, respectively. The number of particles in Section 6.3.2 and Section 6.4.1 are all set to 200 (i.e., $N_{p1} = N_{p2} = 200$).

6.5.2 Influence of Baseline on SLAM System

Recall that the baseline is the separation of two on-board nodes. To evaluate the influence of the baseline on the proposed fusion SLAM, a "hit rate" $\varphi = N_s/N_t$ is defined and evaluated, where N_s is the number of frames that the SLAM system correctly matches the laser points to the already learned map and N_t is the total number of frames. Clearly, a bigger φ corresponds to a better scan matcher. In our evaluation, we compare three SLAM methods: 1) our complete SLAM method,

i.e., UWB localization in Section 6.3 + RASM in Section 6.4, which is denoted as "ours", 2) a simplified version consisting EKF localization (Section IV-A) + RASM, which is denoted as "semi-ours", and 3) the SLAM method proposed in Chapter 5 which only has one on-board UWB node so the length of baseline is 0. For all the methods, the robot is manually pushed alongside a predefined ground truth trajectory three times at the speed of around 0.8m/s to ensure a fair comparison.

The evaluation results are listed in Table 6.2, from which, we can see that a bigger baseline usually results in a higher hit rate φ . We found that success of scan matching (i.e., hit rate φ) strongly depends on the robot's coarse states estimation, and the robot's bearing estimation plays a critical role here. For example, if the length of baseline is comparable with the UWB ranging error, a small vibration of the robot may yield a great change on the robot's bearing estimation. On the other hand, according to the comparison between "ours" and "semi-ours", the hit rate's gap between these two approaches is only 2.1% when the baseline length is 1.5m, while it increases to 10% when the baseline length is about 1.0m. So, using particle filter to further smooth the robot's state estimation proposed in Section 6.3.2 is advised, since a better state estimation of robot is beneficial to the convergence of the followed scan matcher. Finally, the hit rate of "ours" is 11.9% higher than that of Chapter 5. Hence, the proposed scan matcher is more effective than that in Chapter 5, since the robot's coarse states estimation between "ours" and Chapter 5 are similar when baseline length is zero. In the following experiments, the baseline is fixed to 1.0m.

6.5.3 Proposed Fusion SLAM vs. Existing Methods

We compare the proposed SLAM approach in Section 6.2, Section 6.3 and Section 6.4.1 with three state-of-the-art SLAM methods, namely FusionSLAM [193], HectorSLAM [4] and EKF based UWB-only SLAM, in a workshop and a garden scenario. Fig. 6.3 and Fig. 6.4 show the comparison results in workshop and garden, respectively. Our method outperforms [193] in terms of map quality. This

TABLE 6.2: Evaluation on the influence of different baseline.

		baseline length						
		1.5m	1.2m	1.0m	0.8m	0.6m	0.4m	0.0m
Ours	N_s	620	656	610	618	614	553	420
	N_t	829	882	832	868	876	853	836
	$\varphi(\%)$	74.8	74.4	73.3	71.2	70.1	64.8	50.2
Ours _{pf}	N_s	612	592	534	535	530	482	410
	N_t	842	872	839	852	863	869	824
	$\varphi(\%)$	72.7	67.9	63.7	62.8	61.4	55.5	49.8
Chapter 5	N_s	-	-	-	-	-	-	320
	N_t	-	-	-	-	-	-	836
	$\varphi(\%)$	-	-	-	-	-	-	38.3

is because 1) the proposed dual-UWB system provides a better bearing estimation of the robot, 2) no extra noise is added on the laser range measurements in the proposed approach, while [193] adds Gaussian white noise to close up the accuracy gap between UWB ranging and laser ranging. Note that, [193] cannot complete the mapping task in the garden scenario where the node-to-node ranges are mostly NLOS and the robot's location estimation using EKF in [193] suffers big variance. Therefore, the optimization in [193] is easily convergent to a local minimum. In contrast, our proposed SLAM can achieve a satisfactory result because the proposed scan matcher can adaptively change the search region to guide the optimizer to circumvent the local minimals. Finally, the proposed SLAM system is compared with HectorSLAM, a LiDAR-only SLAM which efficiently estimates the odometry through a linearization on the objective function and a multi-resolution grid. The environmental map is constructed upon the estimated odometry. However, the linearization is established upon a small movement of the robot, thus it is prone to accumulate errors when mobile robot is moving fast. This can be seen in Fig. 6.3(e). In contrast, as shown in Fig. 6.3 and Fig. 6.4, the proposed approach does not accumulate error over time, while the map constructed by HectorSLAM, as shown in Fig. 6.4(d), is still noisy even when the operation speed is low.

6.5.4 Accuracy of UWB Beacons' State Estimation vs. Method in Chapter 5

The experiment is conducted in a workshop. The robot is manually pushed alongside a ground truth path three times at speed of around 0.8m/s. Fig. 6.3 and Fig. 6.4 present the estimated robot's states and UWB beacons' states, respectively. Table 6.3 shows the mean and standard deviation of UWB beacons' state estimation errors. From the results, it is clear that UWB beacons' position estimation using the new fusion SLAM is more robust (lower std.) and accurate (smaller mean err.), compared to the method in Chapter 5. It further improves the performance of UWB beacons' states estimation compared to the method in Chapter 5, which demonstrates the effectiveness and robustness of the newly proposed fusion SLAM.

TABLE 6.3: Averaged errors/stds. of five UWB beacons' pose estimates.

	Ours (baseline 1.0m)	Method in Chapter 5	UWB-only
err. in meters	0.072	0.083	0.214
std. in meters	0.075	0.126	0.158

6.6 Conclusion

In this paper, an autonomous exploration system using UWB and LiDAR is presented. We equip two UWB sensors on a robot and deploy several UWB beacon at unknown locations to cover the to-be-explored region. We decouple the SLAM problem into a localization problem and a mapping problem and apply RBPF to solve them sequentially. While solving the UWB-based localization problem, we use EKF to localize all UWB nodes and then apply PF to estimate the robot's state. In the mapping process, we first propose an adaptive-trust-region based recursive scan matching scheme to fine-tune the robot's state based on which the map is updated. One main advantage of fusing UWB and LiDAR for SLAM is that

it avoids the error of the robot's state estimation to accumulate over time because UWB measurements are taken at every time step which helps correct the robot's state. Finally, a fully autonomous exploration scheme by using UWB and LiDAR is proposed to endow the robot the ability of autonomous exploration. Our proposed system is useful for applications such as search and rescue where knowledge of unexplored areas is lack and the mission is time critical.

(a) Proposed SLAM, 0.8m/s

(b) SLAM in Chapter 5, 0.8m/s

(c) UWB-only SLAM 0.8m/s

(d) HectorSLAM 0.4m/s

(e) HectorSLAM 0.8m/s

FIGURE 6.3: Comparison of the proposed fusion SLAM with other SLAM approaches in workshop scenario.

(a) Proposed SLAM, 0.8m/s

(b) SLAM in Chapter 5, 0.8m/s

(c) UWB-only SLAM 0.8m/s

(d) HectorSLAM 0.4m/s

FIGURE 6.4: Comparison of the proposed fusion SLAM with other SLAM approaches in garden scenario.

Chapter 7

Autonomous Exploration Using UWB and LiDAR

7.1 Introduction

Autonomous exploration is one of an important issue which has attracted extensive attentions by researchers, as it has great potential applications among the real world scenarios, such as rescuing the victims. Generally, an autonomous exploration UGV robotic system can be capsulated into three modules in our proposed system as depicted in Fig. 7.1: 1) UWB/LiDAR fusion SLAM (i.e. the approach presented in Chapter 5 and Chapter 6), 2) selecting where-to-explore point (i.e. determining the intermediate points that the robot will be navigated to) and 3) collision-free navigation (i.e. planning a global path and avoiding to colliding with the obstacles). The fusion of UWB and LiDAR happens mainly in two modules: UWB/LiDAR fusion SLAM and selection of where-to-explore point. The former builds a LiDAR map while the robot's location is corrected by UWB measurements and the latter selects a UWB beacon such that there exists a route between itself and the robot's current location that is the least explored. Knowing where-to-explore next, the remaining module is used to enable collision-free navigation.

FIGURE 7.1: The framework of the proposed autonomous exploration system using UWB and LiDAR.

Motivated by the above discussions, we propose a region-aware autonomous exploration robotic system that fuses UWB sensors and a LiDAR sensor for localizing the robot and UWB beacons, collision-free navigation and constructing a map of an unknown environment in real-time. The relation of this chapter to Chapter 5 and Chapter 6 is that the fusion SLAM in Chapter 7 is based on the previous two chapters, but improve the exploration process from manual exploration (Chapter 5 and Chapter 6) to autonomous exploration (Chapter 7). Chapter 7 focuses on the autonomous exploration (how to determine the exploration points for the robot), while Chapter 5 and Chapter 6 focus on the method of fusion SLAM. The key features of the proposed system are highlighted below:

1. For the first time (to the best of our knowledge), UWB and LiDAR are fused for autonomous exploration.
2. We propose to use UWB beacons to cover the region-of-interest where the locations of beacons are estimated on the fly.
3. We consider UWB beacons as robot's intermediate stops and propose a where-to-explore strategy for robot to select the next beacon to move to.
4. The exploration can be done rapidly and the issue of error accumulation has been relieved.

5. The system integrates a motion plan module to enable collision-free exploration.

7.2 Technical Approach

The main framework of the proposed system is shown in Fig. 7.1, which consists of two core modules, namely dual-UWB/LiDAR fusion SLAM and the proposed exploration scheme (including selection of "next-best-view" path, local motion planning). In what follows, we elaborate how we design and implement this exploration scheme.

7.2.1 Dual-UWB/LiDAR Fusion SLAM

As shown in Fig. 7.1, the UWB/LiDAR fusion SLAM is decomposed into the following three coupled steps:

1. estimate the relative positions of the robot and UWB beacons based on UWB range measurements, and derive the robot's heading information,
2. update the LiDAR map based on HectorSLAM [4] while taking into account the UWB measurements,
3. correct robots pose and heading as well as the beacons' poses based on the feedback from the fusion scheme.

Our UWB/LiDAR fusion SLAM module is built based on the one presented in Chapter 5, please refer to Chapter 5 for more details.

In our system, we make two non-trivial modifications from the method in Chapter 5: 1) constructing a dual-UWB configuration where two UWB tags are installed symmetrically on the robot to improve the accuracy of robot's heading estimate; 2) introducing a calibration phase in which an outlier detection scheme is performed

to remove the ranging outliers and a confidence level is assigned to each UWB beacon's location estimation.

7.2.2 Dual-UWB Robotic System

We notice that small error in the robot's heading estimate θ_t may result in large location error of the scan endpoints, especially when scanning the distant objects. In the method presented in Chapter 5, there is only one UWB installed on the robot so that θ_t can only be roughly estimated when the robot is moving, and the heading estimate becomes random when robot comes to a stop. To make the heading estimation more robust, a dual-UWB robot is built in the proposed system. The configuration is shown in Fig. 6.2(a). Due to this modification, the EKF states become $\mathbf{x}_t = \left[\begin{array}{c|c|c} \mathbf{p}_t^{r1} & \mathbf{p}_t^{r2} & \mathbf{p}_t^b \\ \mathbf{v}_t^{r1} & \mathbf{v}_t^{r2} & \mathbf{v}_t^b \end{array} \right] \in \mathbb{R}^{4 \times (N_t+2)}$. The robot's location and velocity are computed by $\mathbf{p}_t^r = (\mathbf{p}_t^{r1} + \mathbf{p}_t^{r2})/2$, $\mathbf{v}_t^r = (\mathbf{v}_t^{r1} + \mathbf{v}_t^{r2})/2$, respectively. Then, (5.1) becomes

$$\begin{aligned} \Delta_{\mathbf{p}_t^r}^*, \Delta_{\theta_t}^*, \Delta_{\mathbf{p}_t^b}^* = & \arg \min_{\Delta_{\mathbf{p}_t^r}, \Delta_{\theta_t}, \Delta_{\mathbf{p}_t^b}} \frac{1}{2} \sum_{i=1}^n [1 - \Gamma(f(\mathbf{p}_t^r + \Delta_{\mathbf{p}_t^r}, \theta_t + \Delta_{\theta_t}, \mathbf{p}_t^{\text{sc}}))]^2 \\ & + \frac{\gamma}{2} \sum_{1 \leq i < j \leq N_t} \left[h(\mathbf{p}_{i,t}^b + \Delta_{\mathbf{p}_{i,t}^b}, \mathbf{p}_{j,t}^b + \Delta_{\mathbf{p}_{j,t}^b}) - \mathbf{r}_t^{\text{UWB}}(k_1) \right]^2 \\ & + \frac{\gamma}{2} \sum_{i=1}^{N_t} \left[h(\mathbf{p}_t^{r1} + \Delta_{\mathbf{p}_t^r}, \mathbf{p}_{i,t}^b + \Delta_{\mathbf{p}_{i,t}^b}) - \mathbf{r}_t^{\text{UWB}}(k_2) \right]^2 + \\ & \left[h(\mathbf{p}_t^{r2} + \Delta_{\mathbf{p}_t^r}, \mathbf{p}_{i,t}^b + \Delta_{\mathbf{p}_{i,t}^b}) - \mathbf{r}_t^{\text{UWB}}(k_3) \right]^2. \end{aligned} \quad (7.1)$$

The last term in (7.1) replaces the last term in (5.1) to account for the dual-UWB setting and $\mathbf{r}_t^{\text{UWB}} \in \mathbb{R}^{(N_t+2)(N_t+1)/2}$.

Vectorizing each of $\mathbf{p}_t^r, \mathbf{p}_t^b, \theta_t$ and stacking them on one another forms a column vector $\boldsymbol{\xi}_t \in \mathbb{R}^{3+2N_t}$. In the same way, $\Delta_{\boldsymbol{\xi}_t} \in \mathbb{R}^{3+2N_t}$ is defined from $\Delta_{\mathbf{p}_t^r}, \Delta_{\mathbf{p}_t^b}, \Delta_{\theta_t}$. To find closed-form solution to (7.1), we approximate functions $\Gamma(f_i(\boldsymbol{\xi}_t + \Delta_{\boldsymbol{\xi}_t}))$

and $h(\boldsymbol{\xi}_t + \boldsymbol{\Delta}_{\boldsymbol{\xi}_t})$, respectively, by their first-order Taylor expansions at point $\boldsymbol{\xi}_t$:

$$\begin{aligned}\Gamma(f_i(\boldsymbol{\xi}_t + \boldsymbol{\Delta}_{\boldsymbol{\xi}_t})) &\approx \Gamma(f_i(\boldsymbol{\xi}_t)) + \Delta_{f_i(\boldsymbol{\xi}_t)}^T \nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t}, \\ h(\boldsymbol{\xi}_t + \boldsymbol{\Delta}_{\boldsymbol{\xi}_t}) &\approx h(\boldsymbol{\xi}_t) + \Delta_{\boldsymbol{\xi}_t}^T \nabla h(\boldsymbol{\xi}_t),\end{aligned}$$

where T denotes transpose operation. Taking derivative of the objective in (7.1) w.r.t. $\boldsymbol{\Delta}_{\boldsymbol{\xi}_t}$ and equating it to zero yields

$$\boldsymbol{\Delta}_{\boldsymbol{\xi}_t}^* = (\boldsymbol{\Pi}_t^{\text{LiD}} - \gamma \boldsymbol{\Pi}_t^{\text{UWB}})^{-1} (\boldsymbol{\Psi}_t^{\text{LiD}} + \gamma \boldsymbol{\Psi}_t^{\text{UWB}}), \quad (7.2)$$

where

$$\begin{aligned}\boldsymbol{\Pi}_t^{\text{LiD}} &= \sum_{i=1}^n \left[\nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t} \right] \left[\nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t} \right]^T, \\ \boldsymbol{\Pi}_t^{\text{UWB}} &= \sum_{1 \leq i < j \leq N_t} \nabla h(\mathbf{p}_{i,t}^b, \mathbf{p}_{j,t}^b) (\nabla h(\mathbf{p}_{i,t}^b, \mathbf{p}_{j,t}^b))^T \\ &\quad + \sum_{i=1}^{N_t} \nabla h(\mathbf{p}_t^{r1}, \mathbf{p}_{i,t}^b) (\nabla h(\mathbf{p}_t^{r1}, \mathbf{p}_{i,t}^b))^T + \\ &\quad \nabla h(\mathbf{p}_t^{r2}, \mathbf{p}_{i,t}^b) (\nabla h(\mathbf{p}_t^{r2}, \mathbf{p}_{i,t}^b))^T \\ \boldsymbol{\Psi}_t^{\text{LiD}} &= \sum_{i=1}^n \left[\nabla \Gamma(f_i(\boldsymbol{\xi}_t)) \frac{\partial f_i(\boldsymbol{\xi}_t)}{\partial \boldsymbol{\xi}_t} \right] [1 - \Gamma(f_i(\boldsymbol{\xi}_t))], \\ \boldsymbol{\Psi}_t^{\text{UWB}} &= \sum_{1 \leq i < j \leq N_t} \nabla h(\mathbf{p}_{i,t}^b, \mathbf{p}_{j,t}^b) (h(\mathbf{p}_{i,t}^b, \mathbf{p}_{j,t}^b) - \mathbf{r}_t^{\text{UWB}}(k_1)) \\ &\quad + \sum_{i=1}^{N_t} \nabla h(\mathbf{p}_t^{r1}, \mathbf{p}_{i,t}^b) (h(\mathbf{p}_t^{r1}, \mathbf{p}_{i,t}^b) - \mathbf{r}_t^{\text{UWB}}(k_2)) + \\ &\quad \nabla h(\mathbf{p}_t^{r2}, \mathbf{p}_{i,t}^b) (h(\mathbf{p}_t^{r2}, \mathbf{p}_{i,t}^b) - \mathbf{r}_t^{\text{UWB}}(k_3)).\end{aligned}$$

Then, we can correct the EKF state χ_t by

$$\mathbf{p}_t^r = \mathbf{p}_t^r + \Delta_{\mathbf{p}_t^r}^*, \quad (7.3)$$

$$\theta_t = \theta_t + \Delta_{\theta_t}^*, \quad (7.4)$$

$$\mathbf{p}_t^{r1} = \mathbf{p}_t^r + \begin{bmatrix} -\frac{L}{2} \sin(\theta_t) \\ \frac{L}{2} \cos(\theta_t) \end{bmatrix}, \quad (7.5)$$

$$\mathbf{p}_t^{r2} = \mathbf{p}_t^r + \begin{bmatrix} \frac{L}{2} \sin(\theta_t) \\ -\frac{L}{2} \cos(\theta_t) \end{bmatrix}, \quad (7.6)$$

$$\mathbf{v}_t^{ri} = \|\mathbf{v}_t^{ri}\| \begin{bmatrix} \cos(\theta_t) \\ \sin(\theta_t) \end{bmatrix}, i = 1, 2, \quad (7.7)$$

$$\mathbf{p}_t^b = \mathbf{p}_t^b + \Delta_{\mathbf{p}_t^b}^*, \quad (7.8)$$

where L is the distance between two fixed UWB sensors equipped on the robot, $\Delta_{\mathbf{p}_t^r}^*, \Delta_{\mathbf{p}_t^b}^*, \Delta_{\theta_t}^*$ are extracted from $\Delta_{\xi_t}^*$.

Update map from $\mathbf{M}_{\rightarrow t-1}$ to $\mathbf{M}_{\rightarrow t}$: In an occupancy grid map, each grid point is assigned an occupancy probability, initially is 0.5 indicating unexplored. When the mapping is on-going, the i -th LiDAR scan endpoint $f_i(\mathbf{p}_{i,t}^{\text{sc}})$ updates the occupancy probability of its nearest grid point located at $\mathbf{p}_{a,b}$ by $\Gamma(\mathbf{p}_{a,b}) = \exp(E_t)/(1 + \exp(E_t))$ where $E_t = E_{t-1} + \text{sign}(U^{\text{occu}}) \log(U^{\text{occu}}/(1 - |U^{\text{occu}}|))$ and $E_0 = 0$. The grid points around the line connecting $\mathbf{p}_{a,b}$ and the robot are considered unoccupied (or free) and their occupancy probabilities are updated in the same manner except replacing U^{occu} by U^{free} . The predefined constants U^{occu} and U^{free} denote the update step length for occupied and free region, respectively. The above process repeats for all the scan endpoints, $i = 1, \dots, n$, observed at time t . The resulting map $\mathbf{M}_{\rightarrow t} = \{\Gamma(\mathbf{p}_{a,b}), \forall a, b\}$ where a, b are the 2D index of the grid points.

Algorithm 3: The proposed autonomous exploration scheme

Require: $\chi_0, \mathbf{M}_{\rightarrow 0}$

```

1:  $t = 0, \mathcal{P}^* = \{\mathcal{P}_i | \max_i \{\|\mathbf{p}_0^r - \mathbf{p}_{i,0}^b\|, i = 1, \dots, N_0\}\}$ 
2: while  $\exists \mathcal{E}_i^{\text{path}} > \mathcal{E}, i = 1, \dots, N_t$  do
3:    $t \leftarrow t + 1$ 
4:    $\mathbf{r}_t^{\text{UWB}} \leftarrow$  UWB measurements at time  $t$ 
5:    $\mathbf{p}_t^{\text{sc}} \leftarrow$  LiDAR scan endpoints at time  $t$ 
6:    $\chi_t \leftarrow \text{EKF}(\mathbf{r}_t^{\text{UWB}}, \chi_{t-1})$ 
7:    $\Delta_\xi^*, \mathbf{M}_{\rightarrow t} \leftarrow \text{Scan matching}(\mathbf{r}_t^{\text{UWB}}, \mathbf{p}_t^r, \mathbf{v}_t^r, \mathbf{p}_t^b, \mathbf{p}_t^{\text{sc}}, \mathbf{M}_{\rightarrow t-1})$  using (7.2)
8:    $\chi_t \leftarrow \text{Correction}(\chi_t, \Delta_\xi^*)$  using (7.3)-(7.8)
9:   if  $\|\mathbf{p}_t^r - \mathbf{p}_{*,t}^b\| \leq \epsilon$  then
10:     $\mathcal{P}_i \leftarrow \text{RRT}(\mathbf{M}_{\rightarrow t}, \mathbf{p}_t^r, \mathbf{p}_{i,t}^b), \forall i = 1, \dots, N_t$  plans a new path from  $\mathbf{p}_t^r$  to  $\mathbf{p}_{i,t}^b$ 
11:     $\mathcal{E}_i^{\text{path}} \leftarrow \text{Entropy}(\mathcal{P}_i)$  using (7.9)
12:     $\mathcal{P}^*, \mathbf{p}_{*,t}^b \leftarrow \mathcal{P}_i$  that maximizes  $\mathcal{E}_i^{\text{path}}$  using (7.10)
13:   end if
14:   if  $\mathcal{P}^*$  is occluded then
15:     $\mathcal{P}^* \leftarrow \text{RRT}(\mathbf{M}_{\rightarrow t}, \mathbf{p}_t^r, \mathbf{p}_{*,t}^b)$ 
16:   end if
17:    $\{v^*, \omega^*\} \leftarrow \text{DWA}(\mathcal{P}^*)$  using (7.11)
18:   robot  $\leftarrow$  send control command  $\{v^*, \omega^*\}$ 
19: end while

```

7.3 Global Path Planning

To guide the robot to approach to the choosing UWB beacon $\mathbf{p}_{i,t}^b$, planning a global path is an effective and necessary way. We implement the rapidly-exploring random tree (RRT) [17] to plan a global trajectory according to the learnt map $\mathbf{M}_{\rightarrow t}$. Starting at the robot's current position, RRT incrementally grows a collision-free tree $\mathcal{T}$. At each iteration, a point is randomly sampled in the map space. Then, RRT attempts to connect it with its nearest vertex in $\mathcal{T}$. If the connection is not occluded by any obstacle, the sample point will be added to $\mathcal{T}$. This process repeats until the distance from the goal point (a location where the robot is planned to move to) to its nearest vertex in $\mathcal{T}$ is less than a predefined value and their connection is not occluded by any obstacle. Then, an unique global path $\mathcal{P} = \{\mathbf{p}_j\}_{j=1}^K$ connecting

the robot to the goal point is established, where $\mathbf{p}_j = [x_j, y_j]^T, j = 1, \dots, K$ are the locations of the vertices in $\mathcal{T}$ on the path and $\mathbf{p}_1 = \mathbf{p}_t^r$.

Remark 8. Since the global path planning is executed under a semi-explored map space, the initially planned path might no longer be a collision-free path as with more information is sensed. Then, a collision-free path to a given destination needs be re-planned when the old one is detected to be occluded by obstacles. Such a process is illustrated in Fig. 7.4(a)-(b).

7.4 Where-To-Explore Path Selection

Exploring in an unknown region, the robot must know where it needs to explore in order to construct a map for the whole region. We propose a where-to-explore path selection strategy that allows the robot to select one beacon (among all UWB beacons) at a time to move to such that the entropy of the selected path from the robot to the beacon is maximized, i.e., the selected path is the least explored.

Using RRT we can construct a path from the robot to each UWB beacon $\mathcal{P}_i = \{\mathbf{p}_j\}_{j=1}^{K_i}, \forall i = 1, \dots, N_t$ where $\mathbf{p}_1 = \mathbf{p}_t^r$ and $\mathbf{p}_{K_i} = \mathbf{p}_{i,t}^b$. Then, we define an entropy for each path [197] as follows

$$\begin{aligned} \mathcal{E}_i^{\text{path}} = & - \sum_{j=1}^{K_i} \sum_{\mathbf{p} \in \mathcal{C}_j} [\Gamma(\mathbf{p}) \log(\Gamma(\mathbf{p})) \\ & + (1 - \Gamma(\mathbf{p})) \log(1 - \Gamma(\mathbf{p}))], \end{aligned} \quad (7.9)$$

where $i = 1, \dots, N_t$, $\mathcal{C}_j = \{\|\mathbf{p} - \mathbf{p}_j\| \leq R\}$ contains the grid points $\mathbf{p} = [x, y]^T$ in distance R from $\mathbf{p}_j$.

Two things we can take away from the definition in (7.9). 1) The entropy of each grid point is maximized at $\Gamma(\mathbf{p}) = 0.5$ indicating unexplored. 2) The further the beacon is away from the robot, the more grid points are evaluated, and thus the

larger the entropy would be. We select the least explored path by

$$\mathcal{P}^* = \arg \max_{\mathcal{P}_i} \mathcal{E}_i^{\text{path}}. \quad (7.10)$$

Let $\mathbf{p}_{*,t}^b$ denotes the location of the selected beacon. If $\|\mathbf{p}_t^r - \mathbf{p}_{*,t}^b\| \leq \epsilon$ where ϵ is a predefined threshold, say 1.5 meter, we start choosing another UWB beacon and planning a new path to it. Furthermore, $\mathcal{E}_i^{\text{path}}$ can be used to terminate the exploration process if $\mathcal{E}_i^{\text{path}} < \mathcal{E}, \forall i = 1, \dots, N_t$, where $\mathcal{E}$ is a predefined threshold, say 10. This means that all the potential paths are all well explored.

7.5 Collision-free Navigation

In the proposed region-aware autonomous exploration system, collision-free navigation is the last and most essential step to enable the robot to safely move among the unknown space, while collecting new information to understand the environment. To navigate the robot safely in the unknown space, a local motion planning approach is implemented which follows the main idea of dynamic window approach (DWA) in [74]. The objective is to select a velocity v and angular velocity ω that brings the robot to the goal with the maximum clearance from any obstacle. Given v, ω , DWA predicts where the robot will be in τ seconds. Assuming v, ω are constants over τ , the predicted location $\mathbf{p}_{t+\tau}^r = \mathbf{p}_t^r + \begin{bmatrix} \frac{v}{\omega} \sin(\theta_t + \omega\tau) - \frac{v}{\omega} \sin(\theta_t) \\ \frac{v}{\omega} \cos(\theta_t) - \frac{v}{\omega} \cos(\theta_t + \omega\tau) \end{bmatrix}$. Given a local goal $\mathbf{p}_j \in \mathcal{P}^*$ (while the global goal is $\mathbf{p}_{i,t}^b \in \mathcal{P}^*$), v and ω are found by

$$\begin{aligned} v^*, \omega^* = \arg \max_{v, \omega} & \alpha |\theta_t + \omega\tau - \angle(\mathbf{p}_j - \mathbf{p}_{t+\tau}^r)| + \\ & \beta \min(\{\|\mathbf{p}_{t+\tau}^r - \mathbf{p}_i^{\text{sc}}\|, i = 1, \dots, n\}) + \\ & \gamma \max(v, 0), \end{aligned} \quad (7.11)$$

where the first term measures the difference between the robot's predicted heading at $t + \tau$ and the heading from the local goal $\mathbf{p}_j$ to the predicted location $\mathbf{p}_{t+\tau}^r$, the second term measures the robot's collision probability at $t + \tau$, and α , β and γ are the weighting factors (in our experiment we choose $\alpha = 0.18, \beta = 0.3, \gamma = 0.2$). We solve (7.11) by an exhaustive search over $v \in [v_t - \Delta_v, v_t + \Delta_v]$ and $\omega \in [\omega_t - \Delta_\omega, \omega_t + \Delta_\omega]$ with step size 0.01, where v_t, ω_t are the robot's current line and angular velocities, respectively. DWA is poor in avoiding moving obstacles as it ignores the obstacles' motion information when solving for v, ω . To address this limitation, we detect true/false for the presence of moving objects based on two consecutive LiDAR scans and stop the robot if a moving obstacle is detected.

The complete exploration scheme including all above mentioned modules is summarized in Algorithm 3.

7.6 Experimental Results

We evaluate our system in two environments: a cluttered indoor workshop and a spacious outdoor garden. The following two comparisons are conducted in our experiments: 1) autonomous exploration vs. exploration with human intervention (i.e. only SLAM module is enabled while the navigation is operated manually); 2) the proposed system vs. one a frontier-based autonomous exploration scheme. To demonstrate the entire exploration process, the experimental tests are recorded with a video available in the supplementary material.

7.6.1 Hardware Platform and Experimental Physical Environment

Fig. 6.2(a) shows a robot of dual-UWB electronic wheelchair that integrates 1) two Decawave UWB sensors which are used for wireless ranging and messaging; 2) a Hokuyo UTM-30LX-EW 2D LiDAR which is used to sense the surrounding

obstacles; 3) two encoders which provide the motion state of the robot, 4) an Arduino Mega board which is used to control the wheelchair; 5) a laptop which runs the core algorithm. Two experimental scenarios (workshop and garden) are shown in Figs. 6.2(b) to 6.2(d).

(a) Workshop scenario

(b) Garden scenario

FIGURE 7.2: Exploration in two different scenarios: indoor and outdoor. Figures from left to right are the maps obtained from the manual exploration (left), the autonomous exploration (middle), and the heat map that shows the difference between these two exploration results (right). Dark green, dark purple and yellow colors in the built map represent unexplored region, explored region and occupied region, respectively. The blue line is the trajectory that the robot has travelled.

7.6.2 Proposed Autonomous Exploration vs. Manual Exploration

We implement the proposed exploration system into two different versions: 1) supervised exploration that the wheelchair is manually pushed to explore the environment, the next-best-view trajectory choosing module and local motion planning module are not implemented in this version. 2) full autonomous exploration which enables all the modules shown in Fig. 7.1. For manual exploration, only the SLAM

module is enabled while the other modules are replaced by human operation. The objective of this experiment is to evaluate the quality of the built map using these two exploration schemes. As an experienced operator is good at avoiding obstacles and finding less explored region, manual exploration is served as the performance benchmark. In Table 7.1, the first two columns under each scenario show our scheme is worse than but close to the manual exploration scheme in terms of the exploration efficiency. The maps learnt by both schemes are shown in Fig. 7.2 from which we can see the qualities of the two maps are very similar.

7.6.3 Proposed Autonomous Exploration System vs. An Existing Autonomous Exploration System

The proposed exploration scheme is also compared with an autonomous exploration system proposed in [5] named Multi-RRT. Fig. 7.3(a) shows that the Multi-RRT scheme operating in a relatively simple environment for a long time is error-prone, while Fig. 7.3(b) shows that the Multi-RRT scheme operation in a complex environment is unable to finish the exploration (it's stuck at $t = 75$ s and struggles for 55 seconds exploring a small region but finally fails). In contrast, our scheme works well in both environments.

The reasons for Multi-RRT's poor performances are 1) there is no means such as a global reference or loop closure detection to eliminate the accumulated error; 2) in the garden detected are many noisy frontier points which easily trap the robot in some regions.

TABLE 7.1: Comparisons among the proposed exploration, manual exploration and Multi-RRT exploration. The * indicates that the exploration is not completed due to the accumulated errors. The + indicates that the result is an average over 20 independent experiments in the same environment.

	workshop			garden		
	Manual	Ours ⁺	Multi-RRT*	Manual	Ours ⁺	Multi-RRT*
time(s)	53	89	90	182	288	fail
trav. dis.(m)	27	34	39	108	121	fail
ave. vel.(m/s)	0.51	0.37	0.42	0.59	0.43	fail

FIGURE 7.3: Comparison of the proposed method and the Multi-RRT exploration scheme [5] under two different scenarios: indoor and outdoor. Figures show the map built at different timestamps.

Remark 9. Generally, UWB has lower range resolution than LiDAR, typically 10cm for UWB ranging error and 1cm for LiDAR ranging error. To balance the discrepancy of the ranging accuracy between two sensor types, we add Gaussian noise with zero mean and standard deviation of 1cm/2cm to the scan endpoints' locations $\mathbf{p}_t^{\text{sc}}$.

7.6.4 Illustration of an Exploration Process in the Garden

Initially, six UWB beacons are placed in the garden at unknown locations. Then, a calibration phase is carried out to localize all the UWB beacons and assign a confidence level to where they locate. Based on the proposed where-to-explore path selection strategy, a path from the robot to UWB beacon 20 is planned as shown in Fig. 7.4(a). The map of the environment is being updated when the robot navigates itself along the planned path. As more information is gathered, the path is re-planned (if necessary) since the previously planned path might be found occluded. The process including path plan, map update and path re-plan is illustrated in Fig. 7.4(b)-(f). When the robot reaches the target point near beacon 20, a next where-to-explore point is determined, which is beacon 22. Then, a path from the robot's current location in Fig. 7.4(f) to beacon 22 is planned. The robot passes through beacon 22, beacon 18 in Fig. 7.4(g) and back to beacon 20 in Fig. 7.4(h). The exploration completes when the entropy of all the where-to-explore paths $\mathcal{E}_i^{\text{path}}, i = 1, \dots, N_t$ are less than a predefined threshold.

FIGURE 7.4: Exploration process in a garden scenario. The yellow circles represent the UWB beacons location estimate while the blue circle represents the robot's location estimate. The red line is the planned path to the selected UWB beacon.

7.6.5 Ambiguous Boundaries

As seen from the area indicated by red circle in Fig. 7.5, there are multiple ambiguous boundaries in the built map. Since a 2D LiDAR is scanning a horizontal plane at a certain vertical level, a small vibration on robot may produce a big influence on the scan endpoints' locations, especially when the distance from the object to the robot is large. The seat in the garden shown in Fig. 7.5 is the protruding part and the LiDAR is almost located at the center of it. Due to the robot's vibration, the laser may hit the area above the seat or beneath the seat thus yielding the ranging inconsistency.

FIGURE 7.5: Ambiguous boundaries when exploring in the garden.

7.7 Conclusion

We have proposed a fully autonomous exploration scheme using UWB and LiDAR, which is built upon the fusion SLAM presented in Chapter 6. Two UWB sensors are mounted on the robot to robustify the robot's states estimation and multiple UWB beacons are deployed to cover the to-be-explored environment. To enable autonomous navigation, the robot selects a UWB beacon at a time as its intermediate stop to move to, and a global path from the robot to a UWB beacon is planned using the RRT method and a collision-free navigation is enabled by the DWA motion planning method. Our proposed system is useful for applications such as search and rescue where knowledge of unexplored areas is lack and the mission is time critical.

Chapter 8

Conclusion and Future Research

8.1 Conclusions

Autonomous UGV technology has plenty of potential applications in recent years. However, it is challenge for a fully autonomous UGV system safely and smartly to fulfill works in various ad-hoc environments, such as cluttered indoor places or the GPS-denied regions. Hence, a satisfactory perception model, a robust SLAM system and a safe collision-free navigation module are desired to enhance the intelligence of the robot and further extend the application of UGVs in complex scenarios. This thesis has focused on two important subtopics of an autonomous UGV, namely visual object tracking and fusion SLAM. Some effective visual object tracking approaches and robust fusion SLAM frameworks have been proposed to handle the challenges in autonomous UGV. The main contributions and conclusions of this thesis are summarized as follows

1. *Event-Triggered tracking in the presence of occlusion and model drift (Chapter 3):*

For visual object tracking, the main challenge hindering improvement is model drift caused by the [challenging](#) attributes like abrupt motion, illumination variation, occlusion, deformation, *et al.* Therefore, identifying the

absence of noisy samples and model drifts is an effective way to enhance the robustness of the tracker. With such motivations, an event-triggered tracking approach with occlusion and drift detection is presented in Chapter 3, which is decomposed into several independent modules involving different subtasks. An event-triggered decision model is designed to coordinate those modules in various scenarios. Moreover, a novel occlusion and drift detection algorithm is developed to tackle the general yet challenging drift and occlusion problems. Experimental evaluations on OTB-100 [2] and VOT16 [3] datasets demonstrate the effectiveness of the proposed tracking framework, which outperforms the state-of-the-art trackers in terms of tracking speed and accuracy.

2. *Adaptive multi-feature reliability re-determination correlation filter for Visual Tracking (Chapter 4):*

The tracker presented in Chapter 3 attempts to alleviate the model drift by detecting and rectifying it. However, due to lack of training samples, it is not an easy task to learn an effective model to identify the model drift. Hence, we propose an online reliability re-determination correlation filter to improve the robustness of the tracker and also alleviate the model drift. Meanwhile, two different evaluation models, named model evaluation and numerical optimization, are proposed and implemented to achieve such an objective. According to experimental evaluations on OTB-50[1], OTB-100[2], TempleColor[33], VOT2016[3] and VOT2018[81], the tracker based on model evaluation is generally more robust and accurate than the numerical optimization based tracker, due to that the former one evaluates the importance of each feature on a similar feature scale. However, the better tracking performance is achieved with the sacrifice of the tracking speed. More importantly, both the proposed two trackers perform favourably against state-of-the-art trackers.

3. *UWB/LiDAR Fusion SLAM via One-step Optimization (Chapter 5):*

Fusion SLAM has found significant applications in robotic system due to its inherent advantages of robustness and efficiency. A fusion scheme that utilizes both UWB and laser ranging measurements for simultaneously localizing the robot, the UWB beacons and constructing the LiDAR map is proposed in Chapter 5. This is a 2D range-only SLAM and no control input is required. In the fusion scheme, a "coarse" map i.e. UWB map is built using UWB peer-to-peer ranging measurements. This "coarse" map can guide where the "fine" map i.e. LiDAR map should be built. Through a new scan matching procedure, the LiDAR map can be constructed while the UWB map is polished. The proposed system suits infrastructure-less and ad-hoc applications where no prior information about the environment is known and the system needs to be deployed quickly. Experimental evaluations conducted in two real scenarios, named cluttered workshop and narrow corridor, have confirmed that the proposed one-step UWB/LiDAR fusion SLAM can accurately build the map for the environment and also be robust to error accumulation.

4. *UWB/LiDAR Fusion SLAM via Step-by-step Iterative Optimization (Chapter 6):*

The fusion scheme proposed in Chapter 5 has two limitations: 1) The closed-form solution is achieved by approximately linearizing the non-linear objective function. Such approximation requires the states variation of robot and UWB beacons be small (i.e. a slow motion of the robot). 2) A composite loss, i.e. sum of UWB loss and LiDAR loss, is formulated to optimize the states of robot and UWB beacons simultaneously. Thus, the trade-off between these two sensors, which has different accuracies, needs to be manually tuned. Hence, an improved fusion scheme is presented in Chapter 6 which refines the state of robot and UWB beacons step-by-step, i.e. firstly refine robot's states using LiDAR range measurements and then rectifies the UWB beacons' states. In detail, two UWB sensors are mounted on a robot, then a RBPF is applied to enhance the robot's bearing estimation and followed by deploying multiple UWB beacons to cover the to-be-explored region. On the

other hand, a coarse-to-fine UWB/LiDAR fusion SLAM framework with an adaptive-trust-region based RBPF is proposed to efficiently find the optimal scan matching. We propose to localize the robot using UWB peer-to-peer ranges measurement and then use LiDAR scan endpoints and the learnt map to refine the robot's states. By doing so, we [eliminate the issue of error accumulation on the robot's state estimation](#). Experimental tests demonstrate that the fusion SLAM suits the infrastructure-free and ad-hoc applications. Meanwhile, no prior information about the environment needs to be prepared.

5. *Autonomous Exploration Using UWB and LiDAR (Chapter 7):*

Autonomous exploration is one important issue which has attracted extensive attentions by researchers. A fully autonomous exploration scheme using UWB and LiDAR is proposed in Chapter 7, which is built upon the fusion SLAM presented in Chapter 6. Two UWB sensors are mounted on the robot to robustify the robot's states estimation and multiple UWB beacons are deployed to cover the to-be-explored environment. To enable autonomous navigation, the robot selects a UWB beacon at a time as its intermediate stop to move to, and a global path from the robot to a UWB beacon is planned using the RRT method and a collision-free navigation is enabled by the DWA motion planning method. The proposed system is useful for applications such as search and rescue where knowledge of unexplored areas is lack and the mission is time critical.

8.2 Recommendations for Future Research

In Chapter 3, one of the most important modules which brings the improvement on the tracking performance is target re-locating. The target re-location model is built upon a traditional machine learning technology (i.e. incremental SVM). However, due to lack of training samples, it is not an easy task to construct a robust yet accurate target detection model. Thus, exploring to construct a more advanced

target detection model is necessary to gain [greater](#) improvement on the tracking performance, for example, deploying the deep learning technology for target re-locating may bring some surprising improvements. However, before implementing such advanced technology, how to prepare large amount yet sufficient samples for model training is an important issue to be addressed.

In Chapter [4](#), two independent solutions are proposed to find the reliability of each feature on the fly. Both the tracking accuracy and robustness have been significantly improved by just fusing two handcraft features and two features from different layers of the VGG-19 [[174](#)] network, which demonstrates that such reliability re-determination is effective. Hence, it is interesting to fuse more [sophisticated](#) features (i.e. ResNet [[198](#)]) to achieve a greater improvement on tracking performance. However, with the increasing of feature types, the tracking speed will be reduced inevitably. Thus, exploring how to make a balance between the number of different features and the tracking speed warrants further investigation and research.

The UWB/LiDAR fusion frameworks presented in Chapter [5](#) and Chapter [6](#) suit for ad-hoc environment. The accuracy of such fusion SLAM highly relies on the robot's location estimation using UWB range measurements. However, UWB sensor can only provide accurate range measurement under the subjection of line-of-sight, otherwise it suffers from the issue of multi-paths and thus reduces the accuracy of location estimation, such as in the scenario shown in Fig. [6.2\(c\)](#) and Fig. [6.2\(d\)](#). Hence, it would be a potential research topic about how to build a robust fusion SLAM system in the presence of noisy UWB range measurements. For example, when the uncertainty of UWB ranges is big, the proposed one-step optimization (i.e. composite loss of UWB and LiDAR) in Chapter [5](#) cannot jump out the local minimal properly, while step-by-step optimization (i.e. RBPF localization + RBPF-ASR scan matching) in Chapter [6](#) might need more iteration steps to find the optimal solution which therefore requires more computational resource. Hence, how to find a method to address great noisy as well as handle the SLAM in real-time would be an interesting research topic.

Chapter 9

Appendix

In this appendix, we provide additional details for the derivations of the proposed numerical optimization and model evaluation based weight solvers, respectively.

9.1 Derivation of numerical optimization based weight solver

Differentiating (4.13) w.r.t. $\mathbf{w}_t, \delta, \lambda_w, \lambda_\delta$ yields

$$\begin{cases} \partial \mathcal{L}_{\mathbf{w}_t} = \frac{1}{2}e_t - \beta \mathbf{w}_t^{-1} + \lambda_w + 2\lambda_\delta(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t}) \\ \partial \mathcal{L}_{\lambda_w} = \sum_{k=1}^L w_{t,k} - 1 \\ \partial \mathcal{L}_{\lambda_\delta} = \varepsilon - \delta - \|\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t}\|_2^2 \\ \partial \mathcal{L}_\delta = -\frac{\beta}{\delta} - \lambda_\delta \end{cases} . \quad (9.1)$$

Next, setting all the derivatives to 0 and rearranging them to yield

$$\begin{cases} \frac{1}{2}e_t \mathbf{w}_t - \beta + \lambda_w \mathbf{w}_t + 2\lambda_\delta(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t})\mathbf{w}_t = 0 \\ \sum_{k=1}^L w_{t,k} - 1 = 0 \\ \varepsilon - \delta - \|\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t}\|_2^2 = 0 \\ -\beta - \lambda_\delta \delta = 0 \end{cases} \quad (9.2)$$

Now the Newton iterative method, i.e., $f(x_0) = -\Delta x \cdot f'(x_0)$, is applied to (9.2) to determine the search directions for $\Delta \mathbf{w}_t, \Delta \lambda_w, \Delta \lambda_\delta$ and $\Delta \delta$, respectively. For example, based on the first equation in (9.2), let $f(\mathbf{w}_t, \delta, \lambda_w, \lambda_\delta) = \frac{1}{2}e_t \mathbf{w}_t - \beta + \lambda_w \mathbf{w}_t + 2\lambda_\delta(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t})\mathbf{w}_t$. Then, using Newton iterative method, we have

$$f = -\Delta \mathbf{w}_t \cdot \frac{\partial f}{\partial \mathbf{w}_t} - \Delta \lambda_w \cdot \frac{\partial f}{\partial \lambda_w} - \Delta \lambda_\delta \cdot \frac{\partial f}{\partial \lambda_\delta} - \Delta \delta \cdot \frac{\partial f}{\partial \delta}.$$

Similarly, we can apply the Newton iterative method to the remaining 3 equations in (9.2). Combining and manipulating all the results, we obtain

$$\begin{bmatrix} \Delta \mathbf{w}_t \\ \Delta \lambda_w \mathbf{I} \\ \Delta \lambda_\delta \mathbf{I} \\ \Delta \delta \mathbf{J}^T \end{bmatrix} = - \begin{bmatrix} \mathbf{H} & \mathbf{Q} & \mathbf{K} & 0 \\ \mathbf{I} & 0 & 0 & 0 \\ \mathbf{G} & 0 & 0 & \mathbf{J} \\ 0 & 0 & \lambda_\delta \mathbf{J}^T & \delta \end{bmatrix}^{-1} \begin{bmatrix} \partial \mathcal{L}_{\mathbf{w}_t} \mathbf{w}_t \\ \partial \mathcal{L}_{\lambda_w} \mathbf{I} \\ \partial \mathcal{L}_{\lambda_\delta} \mathbf{I} \\ \delta \partial \mathcal{L}_\delta \mathbf{J}^T \end{bmatrix} \quad (9.3)$$

where $\mathbf{H} = \frac{1}{2}e_t + 4\lambda_\delta \mathbf{w}_t - 2\lambda_\delta \boldsymbol{\mu}_{t-l:t} + \lambda_w$, $\mathbf{G} = -2(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t})$, $\mathbf{K} = 2(\mathbf{w}_t - \boldsymbol{\mu}_{t-l:t})\mathbf{w}_t$, $\mathbf{Q} = \mathbf{w}_t$, $\mathbf{I} \in \mathbb{R}^{L \times L}$ denotes the identity matrix with size L and $\mathbf{J} \in \mathbb{R}^{L \times 1}$ denotes a vector with ones.

9.2 Derivation of model evaluation based weight solver

The Lagrange formulation of (4.17) can be expressed as

$$\begin{aligned}
\mathcal{L} = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N K_{ij}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) + \sum_{i=1}^N \epsilon(\alpha_i + \alpha_i^*) \\
& - \sum_{i=1}^N y_i(\alpha_i - \alpha_i^*) + \sum_{i=1}^N [\nu_i(\alpha_i - C) + \nu_i^*(\alpha_i^* - C)] \\
& - \sum_{i=1}^N (\eta_i \alpha_i + \eta_i^* \alpha_i^*) + \varphi \sum_{i=1}^N (\alpha_i - \alpha_i^*), \tag{9.4}
\end{aligned}$$

where ν_i , ν_i^* , η_i , η_i^* and φ are the Lagrange multipliers. Optimizing (9.4) leads to the KKT conditions as:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \alpha_i} &= \sum_{j=1}^N K_{ij}(\alpha_j - \alpha_j^*) + \epsilon - y_i + \varphi + \nu_i - \eta_i \\
\frac{\partial \mathcal{L}}{\partial \alpha_i^*} &= - \sum_{j=1}^N K_{ij}(\alpha_j - \alpha_j^*) + \epsilon + y_i - \varphi + \nu_i^* - \eta_i^* \\
\eta_i^{(*)} &\geq 0, \quad \eta_i^{(*)} \alpha_i^{(*)} = 0 \\
\nu_i^{(*)} &\geq 0, \quad \nu_i^{(*)} (\alpha_i^{(*)} - C) = 0. \tag{9.5}
\end{aligned}$$

Then using the defined margin function in (4.19) and KKT conditions in (9.5) yields

$$\begin{cases} h(\mathbf{x}_i) + \epsilon + \nu_i - \eta_i = 0 \\ -h(\mathbf{x}_i) + \epsilon + \nu_i^* - \eta_i^* = 0 \\ \eta_i^{(*)} \geq 0, \quad \eta_i^{(*)} \alpha_i^{(*)} = 0 \\ \nu_i^{(*)} \geq 0, \quad \nu_i^{(*)} (\alpha_i^{(*)} - C) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \tag{9.6}$$

Based on (9.6), we can easily obtain the following conditions for support vectors

$$\begin{aligned} h(\mathbf{x}_i) = \epsilon &\Rightarrow \eta_i > 0, \eta_i^* = \nu_i^* = 0, \\ &\Rightarrow \alpha_i = 0, 0 < \alpha_i^* < C, \Rightarrow 0 > \tau_i > -C. \end{aligned} \quad (9.7)$$

$$\begin{aligned} h(\mathbf{x}_i) = -\epsilon &\Rightarrow \eta_i = \nu_i = 0, \eta_i^* > 0, \\ &\Rightarrow 0 < \alpha_i < C, \alpha_i^* = 0, \Rightarrow 0 < \tau_i < C. \end{aligned} \quad (9.8)$$

Thus, we can find the support set $\mathcal{S}$ which locates at the margin of $h(\mathbf{x}_i)$, that is $|h(\mathbf{x}_i)| = \epsilon$. Therefore $\mathcal{S} = \{i | 0 < |\tau_i| < C\}$.

Whenever a new sample $\mathbf{x}_c$ is added to the training set, the support set $\mathcal{S}$ needs to be adjusted accordingly to ensure that KKT conditions are met. The incremental adjustments on $\Delta\tau_i, i \in \mathcal{S}$ and Δb cause a variation of the margin function as given below

$$\Delta h(\mathbf{x}_i) = K_{ic}\Delta\tau_c + \sum_{j=1}^N K_{ij}\Delta\tau_j + \Delta b. \quad (9.9)$$

Meanwhile, τ_i should also meet the constraint given in (4.17),

$$\tau_c + \sum_{i=1}^N \tau_i = 0. \quad (9.10)$$

Therefore, if $i \in \mathcal{S}$, then $\Delta h(\mathbf{x}_i) = 0$, thus we have

$$\begin{cases} \sum_{j \in \mathcal{S}} K_{ij}\Delta\tau_j + \Delta b = -K_{ic}\Delta\tau_c, i \in \mathcal{S} \\ \sum_{i \in \mathcal{S}} \Delta\tau_i = -\Delta\tau_c. \end{cases} \quad (9.11)$$

Denoting $\mathcal{S} = \{s_1, s_2, \dots, s_l\}$, where l is the number of support vectors, we have

$$\begin{cases} K_{s_1j} \Delta \tau_j + \Delta b &= -K_{s_1c} \Delta \tau_c \\ K_{s_2j} \Delta \tau_j + \Delta b &= -K_{s_2c} \Delta \tau_c \\ &\vdots \\ K_{s_lj} \Delta \tau_j + \Delta b &= -K_{s_lc} \Delta \tau_c \end{cases} \quad (9.12)$$

Thus, (9.11) can be expressed in matrix form as

$$\begin{bmatrix} \Delta b \\ \Delta \tau_{s_1} \\ \vdots \\ \Delta \tau_{s_l} \end{bmatrix} = - \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{s_1s_1} & \cdots & K_{s_1s_l} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{s_ls_1} & \cdots & K_{s_ls_l} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ K_{s_1c} \\ \vdots \\ K_{s_lc} \end{bmatrix} \Delta \tau_c,$$

Author's Publications¹

- **Mingyang Guan**, Changyun Wen, Mao Shan, Cheng-Leong Ng, and Ying Zou, "Real-Time Event-Triggered Object Tracking in the Presence of Model Drift and Occlusion", *IEEE Trans. Industrial Electronics*, vol. 66, no. 3, 2019, pp. 2054-2065.
- **Mingyang Guan** and Changyun Wen, "Adaptive Multi-feature Reliability Re-determination Correlation Filter for Visual Tracking", **Accepted** by *IEEE Transactions on Multimedia*.
- Renjie He, **Mingyang Guan**, Changyun Wen, "SCENS: Simultaneous Contrast Enhancement and Noise Suppression for Low-light Images", **Accepted** by *IEEE Transactions on Industrial Electronics*.
- Yang Song*, **Mingyang Guan***, Wee Peng Tay, Choi Look Law, and Changyun Wen, "UWB/LiDAR Fusion For Cooperative Range-Only SLAM", *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, 2019, pp. 6568-6574.
- **Mingyang Guan**, Changyun Wen, Zhe Wei, Cheng-Leong Ng, and Ying Zou, "A Dynamic Window Approach with Collision Suppression Cone for Avoidance of Moving Obstacles", *IEEE International Conference on Industrial Informatics (INDIN)*, Porto, 2018, pp. 337-342.
- **Mingyang Guan**, Changyun Wen, Kwang-Yong Lim, Mao Shan, Paul Tan, Cheng-Leong Ng, and Ying Zou, "Visual tracking via random partition image

¹The superscript * indicates joint first authors

hashing", *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Phuket, 2016, pp. 1-6.

- Mao Shan, Ying Zou, **Mingyang Guan**, Changyun Wen, and Cheng-Leong Ng, "A leader-following approach based on probabilistic trajectory estimation and virtual train model", *IEEE International Conference on Intelligent Transportation (ITSC)*, Yokohama, 2017, pp. 1-6.

Bibliography

- [1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Comput. Vision and Pattern Recognition*, pages 2411–2418, 2013. [xviii](#), [19](#), [58](#), [59](#), [60](#), [61](#), [81](#), [148](#)
- [2] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1834–1848, 2015. [xviii](#), [xxii](#), [xxv](#), [4](#), [6](#), [8](#), [19](#), [25](#), [27](#), [41](#), [58](#), [59](#), [60](#), [61](#), [62](#), [64](#), [65](#), [68](#), [78](#), [81](#), [148](#)
- [3] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomas Vojir, Gustav Häger, Alan Lukežič, and Gustavo Fernandez. The visual object tracking vot2016 challenge results. Springer, Oct 2016. [xviii](#), [4](#), [19](#), [25](#), [27](#), [63](#), [64](#), [65](#), [68](#), [78](#), [86](#), [148](#)
- [4] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable SLAM system with full 3D motion estimation. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 155–160, 2011. [xxiii](#), [10](#), [31](#), [99](#), [100](#), [101](#), [107](#), [108](#), [117](#), [125](#), [133](#)
- [5] Hassan Umari and Shayok Mukhopadhyay. Autonomous robotic exploration based on multiple rapidly-exploring randomized trees. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1396–1402, 2017. [xxiii](#), [16](#), [35](#), [142](#), [143](#)
- [6] Ulrich Nehmzow. Mobile robotics: Research, applications and challenges. *Future Trends in Robotics. Institution of Mechanical Engineers*, 2001. [1](#)
- [7] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. An affective guide robot in a shopping mall. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 173–180. ACM, 2009. [1](#)
- [8] Seohyun Jeon and Jaeyeon Lee. Multi-robot multi-task allocation for hospital logistics. In *Advanced Communication Technology (ICACT), 2016 18th International Conference on*, pages 339–341. IEEE, 2016. [1](#)
- [9] Rudolph Triebel, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers,

- Michelangelo Fiore, et al. Spencer: A socially aware service robot for passenger guidance and help in busy airports. In *Field and service robotics*, pages 607–622. Springer, 2016. 1
- [10] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 3, 16
- [11] Anthony Stentz and Martial Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2):127–145, 1995. 3, 16
- [12] Sven Koenig and Maxim Likhachev. Improved fast replanning for robot navigation in unknown terrain. In *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, volume 1, pages 968–975. IEEE, 2002. 3, 16
- [13] Mahdi Fakoor, Amirreza Kosari, and Mohsen Jafarzadeh. Humanoid robot path planning with fuzzy markov decision processes. *Journal of applied research and technology*, 14(5):300–310, 2016. 3
- [14] Steven A Wilmarth, Nancy M Amato, and Peter F Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1024–1031. IEEE, 1999. 3
- [15] Charles W Warren. Global path planning using artificial potential fields. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 316–321. Ieee, 1989. 3
- [16] Elon Rimon and Daniel E Koditschek. Exact robot navigation using artificial potential functions. *Departmental Papers (ESE)*, page 323, 1992. 3
- [17] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998. 3, 16, 137
- [18] Jongwoo Kim and James P Ostrowski. Motion planning a aerial robot using rapidly-exploring random trees with dynamic constraints. In *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, volume 2, pages 2200–2205. IEEE, 2003. 3, 16
- [19] Lei Tai, Giuseppe Paolo, and Ming Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36. IEEE, 2017. 3
- [20] Huili Yu, Randal W Beard, Matthew Argyle, and Caleb Chamberlain. Probabilistic path planning for cooperative target tracking using aerial and ground vehicles. In *Proceedings of the 2011 American Control Conference*, pages 4673–4678. IEEE, 2011. 3

- [21] Richard T Vaughan, Gaurav S Sukhatme, Francisco J Mesa-Martinez, and James F Montgomery. Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. In *Distributed autonomous robotic systems 4*, pages 315–324. Springer, 2000.
- [22] Max Bajracharya, Baback Moghaddam, Andrew Howard, Shane Brennan, and Larry H Matthies. A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle. *The International Journal of Robotics Research*, 28(11-12):1466–1485, 2009.
- [23] Carol Cheung and Benjamin Grocholsky. Uav-ugv collaboration with a pack-bot ugv and raven suav for pursuit and tracking of a dynamic target. In *Unmanned Systems Technology X*, volume 6962, page 696216. International Society for Optics and Photonics, 2008. [3](#)
- [24] Marin Kobilarov, Gaurav Sukhatme, Jeff Hyams, and Parag Batavia. People tracking and following with mobile robot using an omnidirectional camera and a laser. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 557–562. IEEE, 2006. [3](#)
- [25] Emina Petrović, Adrian Leu, Danijela Ristić-Durrant, and Vlastimir Nikolić. Stereo vision-based human tracking for robotic follower. *International Journal of Advanced Robotic Systems*, 10(5):230, 2013.
- [26] Rachel Gockley, Jodi Forlizzi, and Reid Simmons. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 17–24. ACM, 2007.
- [27] Guillaume Doisy, Aleksandar Jevtic, Eric Lucet, and Yael Edan. Adaptive person-following algorithm based on depth images and mapping. In *Proc. of the IROS Workshop on Robot Motion Planning*, volume 20, 2012.
- [28] Md Jahidul Islam, Jungseok Hong, and Junaed Sattar. Person following by autonomous robots: A categorical overview. *arXiv preprint arXiv:1803.08202*, 2018. [3](#)
- [29] Michael Strecke and Jorg Stuckler. Em-fusion: Dynamic object-level slam with probabilistic data association. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5865–5874, 2019. [3](#)
- [30] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.
- [31] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019.

- [32] Weichen Dai, Yu Zhang, Ping Li, Zheng Fang, and Sebastian Scherer. Rgb-d slam in dynamic environments using point correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3
- [33] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, 24(12):5630–5644, 2015. 4, 8, 19, 25, 68, 78, 84, 148
- [34] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015. 5, 24, 42, 61, 63, 68
- [35] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Comput. Vision and Pattern Recognition*, pages 2544–2550. IEEE, 2010. 24
- [36] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip Torr. Staple: Complementary learners for real-time tracking. *arXiv preprint arXiv:1512.01355*, 2015. 7, 24, 29, 41, 42, 61, 63
- [37] Xiaoqin Zhang, Weiming Hu, Shengyong Chen, and Steve Maybank. Graph-embedding-based learning for robust object tracking. *IEEE Transactions on Industrial Electronics*, 61(2):1072–1084, 2014.
- [38] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Mach. Vision Conf.*, 2014. 24, 42, 57, 59, 60, 61, 63, 70, 80
- [39] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. Struck: Structured output tracking with kernels. *TPAMI*, 38(10):2096–2109, 2016. 7, 23, 24, 41, 61
- [40] Huihui Song, Yuhui Zheng, and Kaihua Zhang. Robust visual tracking via self-similarity learning. *Electronics Letters*, 53(1):20–22, 2016.
- [41] Kaihua Zhang, Qingshan Liu, Jian Yang, and Ming-Hsuan Yang. Visual tracking via boolean map representations. *Pattern Recognition*, 81:147 – 160, 2018. ISSN 0031-3203. 5
- [42] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012. 5, 23, 24, 42, 61
- [43] Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, and Dacheng Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Comput. Vision and Pattern Recognition*, pages 749–758, 2015. 5, 7, 28, 41, 42, 61
- [44] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term correlation tracking. In *Comput. Vision and Pattern Recognition*, pages 5388–5396, 2015. 6, 7, 8, 24, 27, 28, 41, 42, 61

- [45] Xingping Dong, Jianbing Shen, Dajiang Yu, Wenguan Wang, Jianhong Liu, and Hua Huang. Occlusion-aware real-time object tracking. *IEEE Transactions on Multimedia*, 19(4):763–771, 2017. [6](#), [8](#), [28](#), [42](#), [61](#)
- [46] Xiaofeng Wang and Michael D Lemmon. Event-triggering in distributed networked control systems. *IEEE Transactions on Automatic Control*, 56(3):586–601, 2011. [6](#), [39](#)
- [47] Lantao Xing, Changyun Wen, Zhitao Liu, Hongye Su, and Jianping Cai. Event-triggered adaptive control for a class of uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 2016. [6](#), [39](#)
- [48] Jianming Zhang, Shugao Ma, and Stan Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *European Conf. on Comput. Vision*, pages 188–203. Springer, 2014. [7](#), [29](#), [41](#), [61](#)
- [49] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 4310–4318, 2015. [7](#), [25](#), [41](#), [61](#), [63](#)
- [50] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, pages 6638–6646, 2017. [7](#), [8](#), [26](#), [28](#), [82](#), [84](#), [85](#), [86](#), [87](#)
- [51] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. [7](#), [26](#), [27](#), [82](#)
- [52] Mingyang Guan, Changyun Wen, Mao Shan, Cheng-Leong Ng, and Ying Zou. Real-time event-triggered object tracking in the presence of model drift and occlusion. *IEEE Transactions on Industrial Electronics*, 66(3):2054–2065, 2018. [8](#), [28](#)
- [53] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4904–4913, 2018. [8](#), [25](#), [28](#), [85](#)
- [54] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. [8](#)
- [55] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, pages 4844–4853, 2018. [8](#), [26](#), [29](#), [82](#), [84](#), [86](#), [87](#)
- [56] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2432–2437, 2005. [10](#), [31](#)

- [57] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2D LIDAR SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016. [11](#)
- [58] Chen Wang, Handuo Zhang, Thien-Minh Nguyen, and Lihua Xie. Ultra-wideband aided fast localization and mapping system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1602–1609, 2017. [11](#), [33](#), [34](#)
- [59] Geert-Jan M Kruijff, Fiora Pirri, Mario Gianni, Panagiotis Papadakis, Matia Pizzoli, Arnab Sinha, Viatcheslav Tretyakov, Thorsten Linder, Emanuele Pianese, Salvatore Corrao, et al. Rescue robots at earthquake-hit mirandola, italy: A field report. *IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pages 1–8, 2012. [15](#)
- [60] Jeffrey Delmerico, Elias Mueggler, Julia Nitsch, and Davide Scaramuzza. Active autonomous aerial exploration for ground robot path planning. *IEEE Robotics and Automation Letters*, 2(2):664–671, 2017. [15](#), [36](#)
- [61] Cl Connolly. The determination of next best views. *IEEE International Conference on Robotics and Automation*, 2:432–435, 1985. [15](#)
- [62] Robbie Shade and Paul Newman. Choosing where to go: Complete 3d exploration with stereo. *IEEE International Conference on Robotics and Automation*, pages 2806–2811, 2011.
- [63] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon” next-best-view” planner for 3d exploration. *IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468, 2016. [15](#)
- [64] Farzad Niroui, Kaicheng Zhang, Zendai Kashino, and Goldie Nejat. Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics and Automation Letters*, 4(2):610–617, 2019. [15](#)
- [65] Shi Bai, Fanfei Chen, and Brendan Englot. Toward autonomous mapping and exploration for mobile robots through deep supervised learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2379–2384, 2017. [15](#)
- [66] Brian Yamauchi. A frontier-based approach for autonomous exploration. *cira*, 97:146, 1997. [16](#), [35](#)
- [67] Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. Evaluating the efficiency of frontier-based exploration strategies. *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8, 2010. [16](#), [35](#)

- [68] PGCN Senarathne, Danwei Wang, Zhuping Wang, and Qijun Chen. Efficient frontier detection and management for robot exploration. *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, pages 114–119, 2013. [16](#), [35](#)
- [69] Stefan Oßwald, Maren Bennewitz, Wolfram Burgard, and Cyrill Stachniss. Speeding-up robot exploration by exploiting background information. *IEEE Robotics and Automation Letters*, 1(2):716–723, 2016. [16](#), [36](#)
- [70] Ruth Schulz, Ben Talbot, Obadiah Lam, Feras Dayoub, Peter Corke, Ben Upcroft, and Gordon Wyeth. Robot navigation using human cues: A robot navigation system for symbolic goal-directed exploration. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1100–1105, 2015. [16](#), [36](#)
- [71] K. M. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1160–1165, 2008. ISSN 2153-0858. [16](#)
- [72] Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. In *Ramsete*, pages 181–226. Springer, 2001. [17](#)
- [73] Mišel Brezak, Ivan Petrović, and Nedjeljko Perić. Experimental comparison of trajectory tracking algorithms for nonholonomic mobile robots. In *2009 35th Annual Conference of IEEE Industrial Electronics*, pages 2229–2234. IEEE, 2009. [17](#)
- [74] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997. [17](#), [139](#)
- [75] Oliver Brock and Oussama Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 341–346. IEEE, 1999. [17](#)
- [76] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998. [17](#)
- [77] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, 2008. [17](#)
- [78] Ioannis Karamouzas, Brian Skinner, and Stephen J Guy. Universal power law governing pedestrian interactions. *Physical review letters*, 113(23):238701, 2014. [17](#)

- [79] Zahra Forootaninia, Ioannis Karamouzas, and Rahul Narain. Uncertainty models for ttc-based collision-avoidance. In *Robotics: Science and Systems*, volume 7, 2017.
- [80] Lei Wang, Zhengguo Li, Changyun Wen, Renjie He, and Fanghong Guo. Reciprocal collision avoidance for nonholonomic mobile robots. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 371–376. IEEE, 2018. [17](#)
- [81] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. [19](#), [25](#), [68](#), [78](#), [87](#), [148](#)
- [82] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1442–1468, 2014. [23](#)
- [83] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1619–1632, 2011. [23](#)
- [84] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012. [24](#)
- [85] Ming Tang and Jiayi Feng. Multi-kernel correlation filter for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3038–3046, 2015. [25](#)
- [86] Si Liu, Tianzhu Zhang, Xiaochun Cao, and Changsheng Xu. Structural correlation filter for robust visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4320, 2016. [25](#)
- [87] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *CVPR*, volume 2, page 6, 2017. [25](#), [82](#), [84](#), [85](#)
- [88] Yao Sui, Guanghui Wang, and Li Zhang. Correlation filter learning toward peak strength for visual tracking. *IEEE transactions on cybernetics*, 48(4): 1290–1303, 2017. [25](#)
- [89] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015. [25](#), [26](#), [82](#), [84](#), [86](#), [87](#)
- [90] Kenan Dai, Dong Wang, Huchuan Lu, Chong Sun, and Jianhua Li. Visual tracking via adaptive spatially-regularized correlation filters. In *CVPR*, pages 4670–4679, 2019. [25](#)

- [91] Ming Tang, Bin Yu, Fan Zhang, and Jinqiao Wang. High-speed tracking with multi-kernel correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4874–4883, 2018. [25](#)
- [92] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, volume 3, page 4, 2017. [25](#), [82](#), [84](#), [85](#)
- [93] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6309–6318, 2017. [25](#), [82](#), [85](#), [86](#), [87](#)
- [94] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Int. Conf. Comput. Vision*, 2015. [26](#)
- [95] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Comput. Vision and Pattern Recognition*, pages 4293–4302. IEEE, 2016. [26](#), [86](#)
- [96] Shi Pu, Yibing Song, Chao Ma, Honggang Zhang, and Ming-Hsuan Yang. Deep attentive tracking via reciprocative learning. In *Advances in Neural Information Processing Systems*, pages 1931–1941, 2018. [26](#), [86](#)
- [97] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Int. Conf. Comput. Vision*, pages 3074–3082, 2015. [26](#)
- [98] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488. Springer, 2016. [26](#), [82](#), [84](#), [86](#), [87](#)
- [99] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han. Real-time mdnet. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 83–98, 2018. [26](#)
- [100] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Comput. Vision and Pattern Recognition*, pages 1420–1429. IEEE, 2016. [26](#)
- [101] Xiao Wang, Chenglong Li, Bin Luo, and Jin Tang. Sint++: robust visual tracking via adversarial positive instance generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4864–4873, 2018. [26](#)
- [102] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. [26](#), [63](#), [87](#)

- [103] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1763–1771, 2017. [27](#), [85](#), [87](#)
- [104] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4834–4843, 2018. [27](#)
- [105] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4854–4863, 2018. [27](#)
- [106] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. [27](#)
- [107] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 459–474, 2018. [27](#)
- [108] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 351–366, 2018. [27](#)
- [109] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 101–117, 2018. [27](#), [86](#)
- [110] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2805–2813, 2017. [27](#), [85](#)
- [111] Jongwon Choi, Hyung Jin Chang, Tobias Fischer, Sangdoo Yun, Kyuewan Lee, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Context-aware deep feature compression for high-speed visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 479–488, 2018. [27](#), [82](#), [85](#), [87](#)
- [112] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *European Conf. on Comput. Vision*, pages 749–765. Springer, 2016. [27](#)
- [113] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time object tracking via online discriminative feature selection. *IEEE Transactions on Image Processing*, 22(12):4664–4677, 2013. [28](#)

- [114] Xun Yang, Meng Wang, Luming Zhang, Fuming Sun, Richang Hong, and Meibin Qi. An efficient tracking system by orthogonalized templates. *IEEE Transactions on Industrial Electronics*, 63(5):3187–3197, 2016. [28](#)
- [115] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Int. Conf. Comput. Vision*, pages 4705–4713, 2015. [28](#), [42](#)
- [116] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Learning adaptive correlation filters with long-term and short-term memory for object tracking. *International Journal of Computer Vision (IJCV)*, 2018. [28](#), [61](#)
- [117] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, pages 4335–4343, 2017. [29](#), [82](#), [84](#)
- [118] Naiyan Wang and Dit-Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *ICML*, pages 1107–1115, 2014. [29](#)
- [119] Jiatong Li, Chenwei Deng, Richard Yi Da Xu, Dacheng Tao, and Baojun Zhao. Robust object tracking with discrete graph-based multiple experts. *TIP*, 26(6):2736–2750, 2017. [29](#)
- [120] Dae-Youn Lee, Jae-Young Sim, and Chang-Su Kim. Multihypothesis trajectory analysis for robust visual tracking. In *CVPR*, pages 5088–5096, 2015. [29](#)
- [121] Joseph Djugash, Sanjiv Singh, George Kantor, and Wei Zhang. Range-only SLAM for robots operating cooperatively with sensor networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2078–2084, 2006. [30](#)
- [122] Jose-Luis Blanco, Javier Gonzalez, and Juan-Antonio Fernandez-Madrigal. A pure probabilistic approach to range-only SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2008. [30](#), [31](#)
- [123] Tobias Deißler and Jorn Thielecke. UWB SLAM with Rao-Blackwellized Monte Carlo data association. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010. [30](#)
- [124] Guillem Vallicrosa, Pere Ridao, David Ribas, and Albert Palomer. Active range-only beacon localization for AUV homing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2286–2291, 2014.
- [125] F. Herranz, A. Llamazares, E. Molinos, and M. Ocaña. A comparison of SLAM algorithms with range only sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4606–4611, 2014.

- [126] Lionel Gnev, Olivier Kermorgant, and douard Laroche. A composite beacon initialization for EKF range-only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1342–1348, 2015.
- [127] Janis Tiemann, Andrew Ramsey, and Christian Wietfeld. Enhanced UAV indoor navigation through SLAM-augmented UWB localization. In *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018. [33](#)
- [128] Christian Gentner, Markus Ulmschneider, and Thomas Jost. Cooperative simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope. In *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018. [30](#)
- [129] Joseph Djugash and Sanjiv Singh. A robust method of localization and mapping using only range. *Experimental Robotics*, 54:341–351, Jul. 2008. [30](#)
- [130] Emanuele Menegatti, Andrea Zanella, Stefano Zilli, Francesco Zorzi, and Enrico Pagello. Range-only SLAM with a mobile robot and a wireless sensor networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009. [31](#)
- [131] Karl Granstrm, Thomas B Schn, Juan I Nieto, and Fabio T Ramos. Learning to close loops from range data. *The international journal of robotics research*, 30(14):1728–1754, 2011. [31](#)
- [132] Marian Himstedt, Jan Frost, Sven Hellbach, Hans-Joachim Bhme, and Erik Maehle. Large scale place recognition in 2d lidar scans using geometrical landmark relations. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5030–5035. IEEE, 2014. [31](#)
- [133] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016. [32](#)
- [134] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002. [32](#)
- [135] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. [32](#)
- [136] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. [32](#)

- [137] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013. [32](#)
- [138] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslamlearning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2560–2568, 2018. [33](#)
- [139] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. [33](#)
- [140] Lukas Platinisky, Andrew J Davison, and Stefan Leutenegger. Monocular visual odometry: Sparse joint optimisation or dense alternation? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5126–5133. IEEE, 2017. [32](#), [33](#)
- [141] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006. [32](#)
- [142] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. [32](#)
- [143] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *2012 IEEE International Conference on Robotics and Automation*, pages 1691–1696. IEEE, 2012. [32](#)
- [144] Georg Klein and David Murray. Improving the agility of keyframe-based slam. In *European conference on computer vision*, pages 802–815. Springer, 2008. [32](#)
- [145] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. [32](#)
- [146] Paul Fritsche and Bernardo Wagner. Modeling structure and aerosol concentration with fused radar and LiDAR data in environments with changing visibility. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2685–2690, 2017. [33](#)
- [147] Will Maddern and Paul Newman. Real-time probabilistic fusion of sparse 3D LIDAR and dense stereo. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2181–2188, 2016. [33](#)

- [148] F. J. Perez-Grau, F. Caballero, L. Merino, and A. Viguria. Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and RGB-D sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3495–3502, 2017. [33](#), [34](#)
- [149] Miguel Juliá, Arturo Gil, and Oscar Reinoso. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Autonomous Robots*, 33(4):427–444, 2012. [35](#)
- [150] Farzad Niroui, Ben Sprenger, and Goldie Nejat. Robot exploration in unknown cluttered environments when dealing with uncertainty. *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, pages 224–229, 2017. [35](#)
- [151] Hannah Lehner, Martin J Schuster, Tim Bodenmüller, and Simon Kriegel. Exploration with active loop closing: A trade-off between exploration efficiency and map quality. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6191–6198, 2017. [35](#)
- [152] Boris Sofman, J Andrew Bagnell, Anthony Stentz, and Nicolas Vandapel. Terrain classification from aerial data to support ground vehicle navigation. 2006. [36](#)
- [153] Daniel Perea Ström, Igor Bogoslavskyi, and Cyrill Stachniss. Robust exploration and homing for autonomous robots. *Robotics and Autonomous Systems*, 90:125–135, 2017. [36](#)
- [154] Ben Talbot, Obadiah Lam, Ruth Schulz, Feras Dayoub, Ben Upcroft, and Gordon Wyeth. Find my office: Navigating real space from semantic descriptions. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5782–5787, 2016. [36](#)
- [155] Cyrill Stachniss, O Martinez Mozos, and Wolfram Burgard. Speeding-up multi-robot exploration by considering semantic place information. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1692–1697, 2006. [36](#)
- [156] Shai Shalev-Shwartz and Yoram Singer. Online learning meets optimization in the dual. In *Int. Conf. on Comput. Learn. Theory*, pages 423–437. Springer, 2006. [44](#)
- [157] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27, 2011. [45](#)
- [158] Don Johnson and Sinan Sinanovic. Symmetrizing the kullback-leibler distance. *IEEE Transactions on Information Theory*, 2001. [47](#)
- [159] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. [55](#)

- [160] Jongwon Choi, Hyung Jin Chang, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Visual tracking using attention-modulated disintegration and integration. In *Comput. Vision and Pattern Recognition*, June 2016. 63
- [161] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015. 63
- [162] Zhizhen Chi, Hongyang Li, Huchuan Lu, and Ming-Hsuan Yang. Dual deep network for visual tracking. *IEEE Transactions on Image Processing*, 26(4): 2005–2015, 2017. 63
- [163] Luka Čehovin, Aleš Leonardis, and Matej Kristan. Robust visual tracking using template anchors. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE, 2016. 63
- [164] Tomas Vojir, Jana Noskova, and Jiri Matas. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, 49:250–258, 2014. 63
- [165] Xiaomeng Wang, Michel Valstar, Brais Martinez, Muhammad Haris Khan, and Tony Pridmore. Tric-track: Tracking by regression with incrementally learned cascades. In *Int. Conf. Comput. Vision*, pages 4337–4345, 2015. 63
- [166] Stefan Becker, Sebastian B Krah, Wolfgang Hübner, and Michael Arens. Mad for visual tracker fusion. In *Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII*, volume 9995, page 99950L, 2016. 63
- [167] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer learning based visual tracking with gaussian processes regression. In *European Conf. on Comput. Vision*, pages 188–203. Springer, 2014. 63
- [168] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Comput. Vision and Pattern Recognition*, pages 5374–5383, 2019. 68, 78, 85
- [169] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, pages 1430–1438, 2016. 69
- [170] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, pages 1401–1409, 2016. 70
- [171] Andreas Wächter. *An interior point algorithm for large-scale nonlinear optimization with applications in process engineering*. PhD thesis, PhD thesis, Carnegie Mellon University, 2002. 72
- [172] Francesco Parrella. Online support vector regression. *Master’s Thesis, Department of Information Science, University of Genoa, Italy*, 2007. 75

- [173] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004. [75](#)
- [174] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [78](#), [151](#)
- [175] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, pages 8990–8999, 2018. [82](#), [84](#), [86](#)
- [176] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *ECCV*, pages 353–369, 2018. [82](#), [84](#), [86](#), [87](#)
- [177] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, pages 5486–5494, 2017. [82](#), [84](#), [85](#)
- [178] Xin Li, Chao Ma, Baoyuan Wu, Zhenyu He, and Ming-Hsuan Yang. Target-aware deep tracking. In *CVPR*, pages 1369–1378, 2019. [82](#), [84](#), [86](#)
- [179] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, pages 1308–1317, 2019. [82](#), [84](#), [86](#)
- [180] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4591–4600, 2019. [82](#), [87](#)
- [181] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019. [86](#), [87](#)
- [182] X. R. Li and V. P. Jilkov. A survey of maneuvering target tracking: dynamic models. In *SPIE Conference on Signal and Data Processing of Small Targets*, 2000. [96](#)
- [183] Henk Wymeersch, Jaime Lien, and Moe Z Win. Cooperative localization in wireless networks. *Proceedings of the IEEE*, 97(2):427–450, Feb. 2009. [96](#)
- [184] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z. Win. Ranging with ultrawide bandwidth signals in multipath environments. *Proceedings of the IEEE*, 97(2):404–426, Feb. 2009. [98](#)
- [185] Stefano Marano, Wesley M. Gifford, Henk Wymeersch, and Moe Z. Win. NLOS identification and mitigation for localization based on UWB experimental data. *IEEE Journal on Selected Areas in Communications*, 28(7): 1026–1035, Aug. 2010. [98](#)

- [186] Karthikeyan Gururaj, Anojh Kumaran Rajendra, Yang Song, Choi Look LAW, and Guofa Cai. Real-time identification of NLOS range measurements for enhanced UWB localization. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017. 98
- [187] <https://www.decawave.com/sites/default/files/resources/dwm1000-datasheet-v1.3.pdf>. 98
- [188] Jeroen D. Hol, Fred Dijkstra, Henk Luinge, and Thomas B. Schon. Tightly coupled UWB/IMU pose estimation. In *IEEE International Conference on Ultra-Wideband*, 2009. 98
- [189] Bruce D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence, Vancouver, BC, Canada*, pages 674–679, Aug. 1981. 99
- [190] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking: dynamic models. In *Signal and Data Processing of Small Targets*, volume 4048, pages 212–235. International Society for Optics and Photonics, 2000. 114
- [191] Arnaud Doucet, Nando Freitas, Kevin Murphy, and Stuart Russell. Sequential monte carlo methods in practice. 01 2013. 115
- [192] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005. 115
- [193] Yang Song, Mingyang Guan, Wee Peng Tay, Choi Look Law, and Changyun Wen. UWB/LiDAR fusion for cooperative range-only SLAM. *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 117, 125, 126
- [194] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(12):197–208, 2000. 118
- [195] Nocedal Jorge and Stephen Wright. *Numerical optimization*. Springer Science and Business Media, 2006. 119
- [196] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. In *Sensor devices and systems for robotics*, pages 253–276. Springer, 1989. 120
- [197] Shi Bai, Jinkun Wang, Fanfei Chen, and Brendan Englot. Information-theoretic exploration with bayesian optimization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1816–1822. IEEE, 2016. 138
- [198] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 151