MULTI-SCALE PARTICLE FILTERING FOR MULTIPLE OBJECT TRACKING IN VIDEO SEQUENCES

by

Ahmed Mahmoud

A thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
In partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

ABSTRACT

The tracking of moving objects in video sequences, also known as visual tracking, involves the estimation of positions, and possibly velocities, of these objects. Visual tracking is an important research problem because of its many industrial, biomedical, and security applications. Significant progress has been made on this topic over the last few decades. However, the ability to track objects accurately in video sequences having challenging conditions and unexpected events, e.g., background motion, object shadow, objects with different sizes and contrasts, a sudden change in illumination, partial object camouflage, and low signal-to-noise ratio, remains an important research problem. To address such difficulties, we adopted a multi-scale Bayesian approach to develop robust multiple object trackers.

We introduce a novel concept in the field of visual tracking by adaptively fusing tracking results obtained from a fixed or variable number of wavelet subbands, corresponding to different scene directions and object scales, of a given video frame. Previous approaches to visual tracking were based on using the full- resolution video frame or a smoothed version of it. These approaches have limitations that were overcome by our multi-scale approach that is described in detail in this thesis. This thesis describes the design and implementation of four novel multi-scale visual trackers that are based on particle filtering and the adaptive fusion of subband frames generated using wavelets.

We developed a robust multi-scale visual tracker that represented a captured video frame as different subbands in the wavelet domain. This tracker applied N independent

particle filters to a small subset of wavelet subbands that changed with each captured frame. Then it fused the output tracks of these *N* independent particle filters (tracker fusion) to obtain final position tracks of multiple moving objects in the video sequence.

To reduce the computational cost needed for our first multi-scale visual tracker, we developed a second single cross-section particle filter based robust visual tracker that sequentially fused our adaptively chosen wavelet subbands (data fusion).

We also developed a third robust visual tracker that represented a captured video frame using the Dual-Tree Complex Wavelet Transform (DT-CWT). As most real-valued discrete wavelet transforms suffer from shift variance and low directional selectivity, we used the DT-CWT, instead of a real-valued wavelet transform, to overcome such shortcomings.

Finally, we developed a robust visual tracker that used a cross-section particle filter process a variable number of frame wavelet packet subbands that were adaptively chosen for every video frame (data fusion). These variable number of wavelet packet subbands are equivalent to a sparse representation of a current video frame. The use of wavelet packets, instead of wavelets, for visual tracking is advantageous because it promotes more sparse frame representations, and more directional selectivity for detection of object boundaries.

We evaluated the performance of our novel trackers using different video sequences from the CAVIAR and VISOR databases. Compared to a standard full-resolution particle filter-based tracker, and a single wavelet subband $(LL)_2$ based tracker, our multi-scale trackers demonstrate significantly more accurate tracking performance, in addition to a reduction in average frame processing time.

Acknowledgments

I would like to thank many people who supported me through the preparation of this dissertation. First and foremost, I wish to express my deepest gratitude and sincere thanks to my advisor, Dr. Sherif Sherif, who provided me with a wealth of invaluable information, advice, comments, encouragement, and continuous guidance through this work.

I would also like to thank my examination committee members, Dr. Pourang Irani and Dr. Pradeepa Yahampath, for their additional guidance and support.

Most importantly, I would like to express my special gratitude to my wife, Ola, and my children, Yara and Yassin, for their seemingly endless encouragement, patience, and support. You have been an inspiration and a driving factor in all that I do, and none of this would have been possible without you.

TABLE OF CONTENTS

CHAPTER 1 1

INTRODUCTION 1

1.1 Thesis motivation 2

1.2 Objective 3

1.3 Thesis contributions 3

1.4 Thesis outline 4

CHAPTER 2 6

BACKGROUND AND LITERATURE REVIEW OF BAYESIAN VISUAL TRACKING 6

2.1 Introduction 6

2.2 General approaches to visual tracking 6

2.2.1 Point tracking 7

2.2.2 Kernel tracking 8

2.2.3 Silhouette tracking 9

2.3 Bayesian approach to multiple object tracking 10

2.3.1 Optimal filters 12

2.3.2 Suboptimal filters 16

2.3.3 Literature review for Bayesian visual tracking 26

2.4 Chapter summary 36

CHAPTER 3 38

ROBUST VISUAL TRACKING 38

3.1 Introduction 38

3.2 Multi-scale approach using wavelet domain for robust visual tracking 39

3.2.1 Discrete wavelet transform 39

3.2.2 Bases, frames and linear expansions of signals 39

3.2.3 Scaling functions 41

3.2.4 Wavelet functions 43

3.2.5 Advantages of multiple object tracking in the wavelet domain 44

3.2.6 Review of literature on robust tracking using the wavelet transform 45
3.3 Robust visual tracking using fusion 47
3.4 Robust visual tracking based on machine learning 48
3.4.1 Review of literature on robust tracking using deep-learning 49
3.5 Chapter summary 51

CHAPTER 4 52

ROBUST TRACKING OF MULTIPLE OBJECTS IN VIDEO BY ADAPTIVE FUSION OF SUBBAND PARTICLE FILTERS 52

4.1 Introduction 52
4.2 Implementation of the Adaptive Fusion of Subband PFs multi-scale tracker 54
4.2.1 Initial background extraction and update 54
4.2.2 Generation of subband frames using a multi-scale median transformation 55
4.2.3 Generation of subband difference frames, adaptive subband frame selection 56
4.2.4 Frame denoising 58
4.2.5 Frame binarization and object labeling 59
4.2.6 Implementation of our subband particle filters 61
4.2.7 Fusion of position tracks from our subband particle filters 62
4.2.8 Inter-frame Data association 63
4.3 Performance evaluation of our robust multi-scale visual tracker 64
4.3.1 Example demonstrating partial object camouflage and object shadow 64
4.3.2 Example demonstrating background motion, object shadow, and partial object camouflage 70
4.3.3 Example demonstrating a sudden change in illumination and presence of objects with different sizes 79
4.3.4 Example demonstrating presence of objects with different sizes and partial object camouflage 86
4.3.5 Comparison with correlation filter based visual tracker 89
4.3.6 Practical applicability and average frame processing times 90
4.4 Chapter summary 93

CHAPTER 5 95

ROBUST TRACKING OF MULTIPLE OBJECTS IN VIDEO BY ADAPTIVE FUSION OF *N* FRAME SUBBANDS USING A CROSS-SECTION PARTICLE FILTER 95

5.1 Introduction 95

5.2 Implementation of our cross-section particle filter-based tracker 96

5.2.1 Adaptive selection of subband difference frames 97

5.2.2 Cross-section estimation technique to fuse *N* subband frames 100

5.2.3 Particle filter for fusing *N* subbands based on cross-section estimation technique 101

5.3 Performance evaluation of our cross-section particle filter based tracker 102

5.3.1 Example demonstrating object shadow and partial object camouflage 103

5.3.2 Example demonstrating partial object camouflage and background motion 108

5.3.3 Example demonstrating illumination change, objects of different sizes, and partial object camouflage 115

5.3.4 Example demonstrating presence of objects with different sizes and partial object camouflage 121

5.3.5 Average frame processing times 124

5.4 Chapter summary 125

CHAPTER 6 127

DUAL-TREE COMPLEX WAVELET TRANSFORM FOR ROBUST VISUAL TRACKING 127

6.1 Introduction 127

6.2 Dual-Tree Complex Wavelets 128

6.3 Performance evaluation of our robust DT-CWT based tracker 129

6.3.1 Example demonstrating object shadow, and partial object camouflage 130

6.3.2 Example demonstrating object shadow, and partial object camouflage 135

6.3.3 Example demonstrating a change in illumination and objects of different sizes 142

6.4 Chapter summary 148

CHAPTER 7 150

ROBUST TRACKING OF MULTIPLE OBJECTS IN VIDEO BY ADAPTIVE FUSION OF A VARIABLE NUMBER OF FRAME WAVELET PACKET SUBBANDS 150

7.1 Introduction 150

7.2 Signal denoising using wavelet thresholding techniques 152

7.2.1 Threshold selection 152

7.2.2 Hard thresholding 152

7.3 Wavelet packet transform 153

7.4 *Fast Best Basis Selection* Algorithm 155

7.5 Performance evaluation of our multi-scale WPT based tracker 157

7.5.1 Example demonstrating object shadow and partial object camouflage 157

7.5.2 Example demonstrating background motion, object shadow, and partial object camouflage 163

7.5.3 Example demonstrating the presence of objects with different sizes and sudden illumination change 170

7.5.4 Example demonstrating presence of objects with different sizes and partial object camouflage 177

7.6 Chapter summary 180

CHAPTER 8 182

CONCLUSIONS AND FUTURE WORK 182

8.1 Thesis Contributions 182

8.2 Publications 184

8.3 Possible Future Work 184

REFERENCES 186

APPENDIX A: MULTIRESOLUTION IMAGE PROCESSING 196

1. Image pyramids 196

2. Subband coding 198

3. Haar wavelet transform 199

APPENDIX B: REVIEW OF PROBABILITY DISTRIBUTIONS 200

1. Joint probability 200

2. Statistical independence 201
3. Conditional probability 201

LIST OF FIGURES

Figure 2.1. Main approaches to visual tracking 7

Figure 3.1. Nested spaces spanned by scaling functions 43

Figure 3.2. Difference between two adjacent subspaces is spanned by wavelet functions 43

Figure 4.1. Implementation of our multi-scale visual tracker 53

Figure 4.2. Subband frames from level 1 and 2 used in our tracker 58

Figure 4.3. Result of subband generation and frame binarization for the adaptively chosen subbands $(LL)_1$, $(HL)_1$, and $(HL)_2$ for the 4^{th} frame in *''OneLeaveShopReenter2front"* video sequence. (a) Background, current, and binary frames, respectively, for the $(LL)_1$ subband (upper row); (b) Background, current, and binary frames, respectively, for the $(HL)_1$ subband (middle row); and (c) Background, current, and binary frames, respectively, for $(HL)_2$ subband (bottom row) 60

Figure 4.4. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) a single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 65

Figure 4.5. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 65

Figure 4.6. Binary frames generated from the 265^{th} frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$ 68

Figure 4.7. Visual tracking results for 265^{th} video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 68

Figure 4.8. Binary frames generated from the 100th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$ 70

Figure 4.9. Visual tracking results for 100th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 70

Figure 4.10. Position tracks of true objects (a) - (i) in the *''OneLeaveShopReenter2front"* video using a standard full-resolution particle filter-based tracker (right column), an LL-based tracker (middle column), and our multiscale tracker (left column) 72

Figure 4.11. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our proposed multiscale 73

Figure 4.12. Binary frames generated from the 6th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_2$ 75

Figure 4.13. Visual tracking results for the 6th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 75

Figure 4.14. Binary frames generated from the 116th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$ 76

Figure 4.15. Visual tracking results for the 116th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 77

Figure 4.16. Binary frames generated from the 427th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_2$ 78

Figure 4.17. Visual tracking results for the 427th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 79

Figure 4.18. Position tracks of true objects (a) - (i) in the *"Meet_WalkTogether2"* video using a standard full resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multiscale tracker (left column) 81

Figure 4.19. Position tracks of phantom objects generated by: (a) the standard full resolution particle filter-based tracker, (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 82

Figure 4.20. Binary frames generated from the 56^{th} frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(LL)_1$ 84

Figure 4.21. Visual tracking results for the 56^{th} video frame using: (a) the standard full-resolution particle filter-based tracker; (b)) the single wavelet subband $(LL)_2$ based tracker; and (c) our multi-scale tracker 84

Figure 4.22. Binary frames generated from the 201^{st} frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(LH)_2$ 85

Figure 4.23. Visual tracking results for the 201^{st} video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker 85

Figure 4.24. Visual tracking results for four video frames using our multi-scale tracker 88

Figure 4.25. Position tracks of true objects (a) - (i) in the "*ATCS*" video using a standard full resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multiscale tracker (left column) 88

Figure 5.1. Flowchart of our cross-section particle filter-based tracker 99

Figure 5.2. Cross-section estimation technique (a) current frame time is frozen to process N subband frames; (b) Cross-section estimation over the timeline 100

Figure 5.3. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter-based tracker 104

Figure 5.4. Binary frames generated from the 267th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$ 106

Figure 5.5. Visual tracking results for the 267th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our tracker; and (c) our cross-section particle filter based tracker 106

Figure 5.6. Binary frames generated from the 240th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(HL)_2$ 107

Figure 5.7. Visual tracking results for the 240th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker 107

Figure 5.8. Position tracks of true objects (a) - (f) in the *''OneLeaveShopReenter2front"* video using our multi-scale tracker (upper row), and our cross-section particle filter based tracker (lower row) 109

Figure 5.9. Binary frames generated from the 88th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(HL)_2$ 111

Figure 5.10. Visual tracking results for the 88th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale filter-based tracker; and (c) our cross-section particle filter based tracker 112

Figure 5.11. Binary frames generated from the 116th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(HL)_2$ 113

Figure 5.12. Visual tracking results for the 116th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker 113

Figure 5.13. Binary frames generated from the 138th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(LL)_2$ 114

Figure 5.14. Visual tracking results for the 138th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker 115

Figure 5.15. Position tracks of true objects (a) - (f) in the *"Meet_WalkTogether2"* video using our multi-scale tracker (upper row), and our cross-section particle filter based tracker (lower row)..... 116

Figure 5.15. Binary frames generated from the 67th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(LH)_2$..... 118

Figure 5.16. Visual tracking results for the 67th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker 119

Figure 5.17. Binary frames generated from the 202nd frame using: (a) the full-resolution frame; (b) subband $(LH)_2$; (c) subband $(HL)_2$ 120

Figure 5.18. Visual tracking results for the 202nd video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker 120

Figure 5.19. Visual tracking results for four video frames using our multi-scale tracker 123

Figure 5.20. Position tracks of true objects (a) - (f) in the "*ATCS*" video using our multi-scale tracker (upper row), and our cross-section particle filter based tracker (lower row) ... 123

Figure 6.1. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) a single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 131

Figure 6.2. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker..... 131

Figure 6.3. Binary frames generated from the 100th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) one of the chosen subbands in our multi-scale DT-CWT based tracker 133

Figure 6.4. Visual tracking results for the 100th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 134

Figure 6.5. Binary frames generated from the 267th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) one of the chosen subbands in our multi-scale DT-CWT based tracker 134

Figure 6.6. Visual tracking results for the 267th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 135

Figure 6.7. Position tracks of true objects (a) - (i) in the *''OneLeaveShopReenter2front''* video using a standard full-resolution particle filter-based tracker (right column), an LL-based tracker (middle column), and our multi-scale DT-CWT based tracker (left column).. 137

Figure 6.8. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker; and (c) our multi-scale DT-CWT based tracker..... 138

Figure 6.9. Binary frames generated from the 305th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) one of the chosen subbands in our multi-scale DT-CWT based tracker 140

Figure 6.10. Visual tracking results for the 305th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 140

Figure 6.11. Binary frames generated from the 427th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) a chosen subband by multi-scale DT-CWT based tracker..... 141

Figure 6.12. Visual tracking results for the 427th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 141

Figure 6.13. Position tracks of true objects (a) - (i) in the *"Meet_WalkTogether2"* video using a standard full resolution particle filter-based tracker (right column), a single wavelet subband $(LL)_2$ based tracker (middle column), and our multi-scale DT-CWT based tracker (left column) 143

Figure 6.14. Position tracks of phantom objects generated by: (a) the standard full resolution particle filter-based tracker, (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 144

Figure 6.15. Binary frames generated from the 67th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) a chosen subband by multi-scale DT-CWT based tracker..... 146

Figure 6.16. Visual tracking results for the 67th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 146

Figure 6.17. Binary frames generated from the 200th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) a chosen subband by multi-scale DT-CWT based tracker..... 148

Figure 6.18. Visual tracking results for the 200th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker 148

Figure 7.1. Hard thresholding operation, assuming the input signal was normalized to the range of [-1, 1] and the threshold level was set to τ =0.4: (a) original signal; (b) hard thresholded signal..... 153

Figure 7.2. Quad-tree representing a two-dimension wavelet packet transform 155

Figure 7.4. shows an example of a result obtained by the *Fast Best Basis Selection* algorithm for representing the 10th *difference frame* in the *''OneLeaveShopReenter2front"* video sequence. 157

Figure 7.3. Example of a result obtained from the fast best basis selection algorithm (a) 10th *difference frame* in the *''OneLeaveShopReenter2front"* video sequence represented by its best wavelet packet basis; (b) best wavelet packet basis for this 10th difference frame. 157

Figure 7.5. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) a single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker. 158

Figure 7.6. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker. 159

Figure 7.7. Binary frames generated from the 265th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) a subband frame from the constructed best wavelet packet tree 161

Figure 7.8. Visual tracking results for 265th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker 161

Figure 7.9. Binary frames generated from the 245th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree 162

Figure 7.10. Visual tracking results for 245th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker 162

Figure 7.11. Position tracks of true objects (a) - (i) in the *''OneLeaveShopReenter2front"* video using a standard full-resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multi-scale WPT based tracker (left column) 165

Figure 7.12. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker; and (c) our multi-scale WPT based tracker..... 166

Figure 7.13. Binary frames generated from the 116th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree 168

Figure 7.14. Visual tracking results for the 116th video frame using; (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker..... 168

Figure 7.15. Binary frames generated from the 427th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree 169

Figure 7.16. Visual tracking results for the 427th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker..... 170

Figure 7.17. Position tracks of true objects (a) - (i) in the *"Meet_WalkTogether2"* video using the standard full-resolution particle filter-based tracker (right column), the single wavelet subband $(LL)_2$ based tracker (middle column), and our multi-scale WPT based tracker (left column) 171

Figure 7.18. Position tracks of phantom objects generated by: (a) the standard full resolution particle filter-based tracker, (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker..... 172

Figure 7.19. Binary frames generated from the 45th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree 175

Figure 7.20. Visual tracking results for 45th video frame using: (a) the standard particle filter full resolution based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker..... 175

Figure 7.21. Binary frames generated from the 201st frame using; (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree 176

Figure 7.22. Visual tracking results for the 201st video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker..... 176

Figure 7.23. Visual tracking results for four video frames using our multi-scale tracker 178

Figure 7.24. Position tracks of true objects (a) - (i) in the "*ATCS*" video using a standard full resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multiscale WPT based tracker (left column) 178

LIST OF TABLES

Table 2.1. Standard particle filter using sequential importance resampling 25

Table 4.1. Number of missed object events, average position track errors, and number of phantom object events 67

Table 4.2. Number of missed object events, average position track errors, and number of phantom object events 74

Table 4.3. Number of missed object events, average position track errors, and number of phantom object events 83

Table 4.4. Number of missed object events, average position track errors, and number of phantom object events 89

Table 4.5. Average frame computation times using single wavelet subband $(LL)_2$ based tracker, a standard full- resolution particle filter-based tracker, and our proposed tracker 91

Table 4.6. Computational complexity of our multi-scale N subband particle filters-based tracker 92

Table 5.1. Cross-section particle filter algorithm 101

Table 5.2. Number of missed object events, average position track errors, and number of phantom object events 105

Table 5.3. Number of missed object events, average position track errors, and number of phantom object events 110

Table 5.4. Number of missed object events, average position track errors, and number of phantom object events 117

Table 5.5. Number of missed object events, average position track errors, and number of phantom object events 124

Table 5.6. Average frame computation times using our *N* subband particle filter-based tracker and our proposed sequential particle filter-based tracker 125

Table 6.1. Number of missed object events, average position track errors, and number of phantom object events 132

Table 6.2. Number of missed object events, average position track errors, and number of phantom object events 139

Table 6.3. Number of missed object events, average position track errors, and number of phantom object events 145

Table 7.1. *Fast Best Basis Selection* from *redundant* tree dictionaries 156

Table 7.2. Number of missed object events, average position track errors, and number of phantom object events 160

Table 7.3. Number of missed object events, average position track errors, and number of phantom object events 167

Table 7.4. Number of missed object events, average position track errors, and number of phantom object events 173

Table 7.5. Number of missed object events, average position track errors, and number of phantom object events 180

Chapter 1

Introduction

Tracking of moving objects in video sequences, also known as visual tracking, typically involves estimating the position and velocity of a single object or a group of moving objects. Visual tracking is an important research problem because of its many industrial, biomedical, and security applications. These applications can be grouped into six main areas [1]:

1. Tele-collaboration and interactive gaming: for example, on-desk video conferencing or an Xbox gaming console's Kinect software—wherein standard cameras are augmented with visual tracking software to localize and follow users.
2. Medical applications and biological research: for example, using a visual tracker to estimate the position of particular soft tissues or instruments, such as needles, during surgery.
3. Media production and augmented reality: for example, the use of a camera tracker can enable the addition of computer graphic components and special effects to an originally captured shot by estimating 3D information about the scene, such as the camera's position and orientation, over time.
4. Robotics and unmanned vehicles. For instance, a visual tracker can be used to estimate global motion for environmental exploration and mapping using unmanned aerial vehicles (UAVs).

5. Surveillance and business intelligence. One example of this application is retail intelligence wherein a visual tracker is used to estimate the trajectories of customers in retail places to determine where they spend their time and how they interact with different products.
6. Art installations and performances; for instance, tracking technology that allows museum-goers to interact with visual installations.

In this chapter, Section 1.1 discusses the motivation for this thesis, while Section 1.2 states its objective, and Section 1.3 details its contributions to the development of robust visual trackers. Finally, Section 1.4 outlines the organizational structure of this thesis.

1.1 Thesis motivation

There has been much progress in the development of visual tracking technology over the last few decades. However, robust visual tracking is still an open research problem [2, 3]. Robust visual tracking refers to the ability to track objects accurately in video sequences that are characterized by unexpected events and challenging conditions [2-4]. Such unexpected events and challenging conditions could include:

1. Presence of background motion and object shadows.
2. Presence of objects with different sizes.
3. Presence of objects with different contrast levels.
4. low signal-to-noise (SNR) ratio.
5. Sudden changes in illumination.
6. Presence of partial object camouflage.
7. Real-time processing requirement.

1.2 Objective

The main objective of this thesis is to design and implement robust multiple-object visual trackers that address most of the previously introduced challenges, and that can track moving targets in the FOV (Field of View) of one sensor (video camera).

1.3 Thesis contributions

We utilize a multi-scale Bayesian approach and a Bayesian approach to design and implement robust multiple-object trackers. We introduce a new concept in the field of visual tracking that is the adaptive fusion of a chosen number, N, or even a variable number of wavelet subbands from different directions and scales. The previous related approaches were based on utilizing a full-resolution video frame or lower resolution, i.e., a smoothed version, of a full-resolution video. These approaches have problems that are addressed by our multi-scale approach as discussed in detail in Section 3.2.5.

This thesis describes four novel multi-scale video trackers that are based on particle filtering and adaptive choice of subband frames:

1. A robust multi-scale visual tracker that represented a captured video frame as different subbands in the wavelet domain. This tracker applied N independent particle filters to a small subset of wavelet subbands that changed with each captured frame. Finally, this tracker fused the output tracks of these N independent particle filters to obtain final position tracks of multiple moving objects in the video sequence (tracker-fusion).

2. A robust multi-scale visual tracker that used a single particle filter to fuse our adaptively chosen wavelet subbands (data-fusion) sequentially.
3. A robust visual tracker that represented a captured video frame using the Dual-Tree Complex Wavelet Transform (DT-CWT). As most real-valued discrete wavelet transforms suffer from shift variance and low directional selectivity, we used the DT-CWT, instead of a real-valued wavelet transform, to overcome such shortcomings.
4. A robust visual tracker that used a sparse video frame representation obtained using a *Fast Best Basis Selection* algorithm applied to its wavelet packet tree. A single particle filter sequentially fused the subbands that were adaptively chosen from the wavelet packet tree of the captured video frame (data-fusion). Wavelet packets are advantageous, compared to orthogonal or biorthogonal wavelets, for visual tracking because they allow more *directional selectivity* in the detection of object boundaries.

1.4 Thesis outline

This thesis consists of eight chapters. Chapter 2 provides a detailed background and literature review for Bayesian visual tracking, while Chapter 3 presents the background and literature review for robust visual tracking approaches, including wavelet transform and data fusion. Chapter 4 describes our robust multiple-object visual tracker that uses an adaptive fusion of subband particle filters. Chapter 5 presents our robust multiple-object visual tracker that uses an adaptive sequential fusion of N frame subbands with a single particle filter. Chapter 6 presents our DT-CWT based visual tracker, while Chapter 7

describes the development of our wavelet packet transform based robust visual tracker. Finally, Chapter 8 presents our conclusions and suggestions for future research.

Chapter 2

Background and Literature Review of Bayesian Visual Tracking

2.1 Introduction

Many different techniques for visual tracking have been proposed [5, 6]. Visual tracking can be classified into three categories: point tracking, kernel-based tracking, and silhouette-based tracking [4, 7, 8]. Also, approaches to point tracking can be sub-divided into either deterministic or statistical. In this thesis, we focus on the Bayesian statistical approach for visual tracking. Kalman filters and particle filters have proven to be powerful and reliable methods for visual tracking, and they can be viewed as information fusers [9] due to their ability to combine observational data and a dynamic model for a given object into one mathematical framework. Moreover, this ability can be extended to provide a consistent framework for data fusion [10], which is why they are an attractive option for visual tracking even though their computational cost tends to increase as the number of tracked objects increases [6].

In this chapter, Section 2.2 briefly outlines the general approaches to visual tracking, while Section 2.3 describes the Bayesian approach to multiple-object tracking, including a review of the relevant literature. Finally, Section 2.4 provides a chapter summary.

2.2 General approaches to visual tracking

There are several object-tracking techniques: point tracking, kernel-based tracking, and silhouette-based tracking. Figure 2.1 shows a tree diagram depicting the various

divisions and sub-divisions of visual tracking, which are discussed in the following subsections.

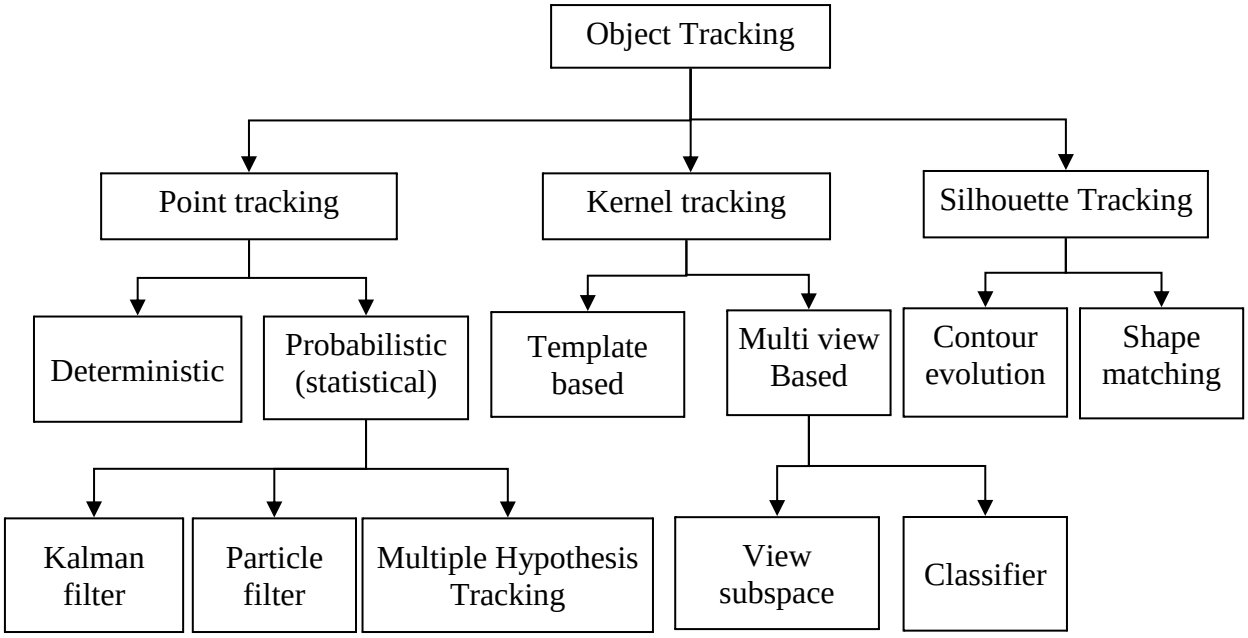


Figure 2.1. Main approaches to visual tracking

2.2.1 Point tracking

In the point tracking approach, objects are represented by points, and tracking can be formulated by finding the correspondence of these points across frames. The correspondence of the object points between consecutive frames is based on the previous object state; namely, the object's position and acceleration. Point tracking can be divided into two approaches [5]:

1. Deterministic tracking.
2. Statistical tracking.

The deterministic tracking approach uses qualitative motion heuristics to constrain the correspondence problem, e.g., the maximum velocity and acceleration of the object. On the other hand, the statistical tracking approach—also known as probabilistic tracking —explicitly takes the uncertainties of measurement and motion model into account to establish point correspondence [4]. Typically, the statistical tracking approach is comprised of a Bayesian framework, e.g., Kalman filters, particle filters, or Multiple Hypothesis Tracking (MHT).

2.2.2 Kernel tracking

In the kernel tracking approach, objects are represented by primitive shapes, such as ellipses or rectangles. Kernel tracking is performed by computing the motion of the object region across frames [5]. The object motion is generally in the form of parametric, e.g., translation, rotation, and affine, motion. Visual tracking techniques using this approach differ depending on the appearance representations or methods used to estimate the object's motion. Kernel tracking can be divided into two approaches based on the selected appearance representation:

1. Template and density-based appearance models.
2. Multi-view appearance models.

In the template and density-based models approach, the appearance model, which could be a simple template or a color histogram model, is usually generated online from the most recent video frames. Using this simple template, visual tracking is performed by a brute force search to find a region in the current video frame that best matches a reference

image template—usually an object region from the previous video frame. In the color histogram model, which is typically generated from an ellipsoidal or rectangle region containing an object, visual tracking can be performed by using a *Mean Shift* method to find the area in the current video frame that provides the color histogram distribution that is most similar to the object color histogram model from previous frame [11].

If object view changes dramatically during tracking, appearance models, such as simple template and color histogram, may no longer be valid and the object track might be lost. Therefore, in the multi-view appearance models approach, different views of the object can be generated offline and used for tracking. Both the eigenvector-based tracker in [12], and the classifier-based tracker using a Support Vector Machine (SVM) [13], are examples of using this approach.

2.2.3 Silhouette tracking

The use of simple geometric shapes, such as rectangles, may not be suitable for representing objects with complex shapes, for example, hands, fingers, and shoulders. Silhouette visual tracking offers accurate shape-representation models for objects with complex shapes [5]. In this approach, visual tracking is performed by finding the object region in the current frame that best matches the object model obtained from the previous frame. There are two approaches for Silhouette tracking:

1. Shape Matching.
2. Contour Tracking.

Shape matching can be performed like the template-matching-based tracking method described in Section 2.2.2. To start visual tracking, an object reference model, or an edge map, is obtained from the previous frame. Once this has been done, a similarity measure is computed between each candidate object region in the current video frame and this reference model. The region with the best match becomes the new object region. To cope with changes in the object's appearance, the reference model is reinitialized with the model obtained from this region.

In the contour tracking approach, an initial contour representing an object in the previous frame iteratively evolves to the new position in the current frame. This contour evolution requires that a portion of the object in the current frame be overlaid with the object region in the previous frame. Contour tracking can be implemented using two different approaches. The first approach uses state-space models to model the contour's shape and motion [14]. The second approach uses direct minimization techniques to directly evolve the contour by minimizing its energy functions [4]. The advantage of silhouette tracking is its adaptability in managing a wide range of different object shapes [5].

2.3 Bayesian approach to multiple object tracking

Visual tracking can be formulated as a sequential Bayesian state estimation problem [10, 15, 16]. The state vector of $n \in \{1, \dots, N\}$ objects to be tracked is represented as $X_i = (x_i^1, x_i^2, \dots, x_i^N)^T$. where x_i^n refers to the state vector of the n^{th} object at discrete-time $i \in \{1, \dots, t\}$, and $(\cdot)^T$ denotes the transpose operator. Typically, x_i^n contains values of kinematic variables such as position, velocity, and acceleration. Sequential Bayesian state

estimation aims to determine the *a posteriori* distribution of the current state vector, X_t, given the range between the first video frame and the current one, $Z_{1:t} = \{Z_1, Z_2, \dots, Z_t\}$. The time evolution of the state vector is described by a discrete-time stochastic model [15], which is given by,

$$X_t = F_t(X_{t-1}, v_{t-1}) \tag{2.1}$$

where $F_t(\cdot)$ represents *a priori* knowledge of the object dynamics and v_{t-1} represents system noise. This *motion model* could also be written as a transition probability density function, $p(X_t|X_{t-1})$. The state vector at time *t* is related to the video frame at time *t* using,

$$Z_t = H_t(X_t, w_t) \tag{2.2}$$

where $H_t(\cdot)$ represents a measurement model and w_t represents measurement noise. This model is typically known as the *likelihood* function. For sequential Bayesian state estimation, a state prediction step and a state update step are required.

The state prediction step uses the object dynamics model, Eq. (2.1), to predict the state vector's evolution from time $t-1$ to time t. Let the *a posteriori* distribution of the state vector $p(X_{t-1}|Z_{1:t-1})$ at time $t-1$ be known, and let the object dynamic model be available as a transition probability density function, $p(X_t|X_{t-1})$. Using the Chapman-Kolmogorov equation, a prediction of the current state vector X_t is given by,

$$p(X_t|Z_{1:t-1}) = \int p(X_t|X_{t-1})\, p(X_{t-1}|Z_{1:t-1})\, dX_{t-1} \tag{2.3}$$

The state update step uses Z_t, which is the frame at time t, to update the estimate of the state vector obtained from the prediction step. Using Bayes' rule, an updated estimate of the current state vector X_t is given by,

$$p(X_t|Z_{1:t}) = \frac{p(Z_t|X_t)\, p(X_t|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \tag{2.4}$$

where the denominator in Eq. (2.4) is an integral that represents a normalizing constant. Both Eq. (2.3) and Eq. (2.4) are considered the foundation of sequential Bayesian state estimation.

Optimal filters, or suboptimal filters, could be used to exactly, or approximately, to solve the integrals in Eq. (2.3) and Eq. (2.4), respectively. Optimal filters and suboptimal algorithms are described in the next sections.

2.3.1 Optimal filters

Under certain assumptions, optimal filters can be employed to obtain the exact solution of a sequential Bayesian state estimation problem. For example, assuming Gaussian statistics and *F* and *H* being linear functions, the Kalman filter becomes the optimal filter for solving the integrals in Eq. (2.3) and Eq. (2.4) ; however, if the state space is discrete and consists of a finite number of states, an optimal solution is the Grid-based method.

2.3.1.1 Kalman filtering

Under highly restrictive assumptions; namely, *linear-Gaussian* assumptions, which entail the assumption of Gaussian statistics and *F* and *H* being linear functions. Under these

assumptions, the Kalman filter can produce an optimal estimate [17], These *linear-Gaussian* assumptions imply the following conditions:

1. The probability distribution of the objects' state vector at any time, t, is Gaussian. Furthermore, the probability distribution of the objects' state vectors for any finite set of times is a multivariate Gaussian distribution,
2. The measurements are linear functions of the objects state vector,
3. The measurent errors are represented as additive Gaussian noise, and are independent of the objects' state at the time of the measurement, and independent of the measurement errors at all other times.

Therefore, *a posteriori* distribution at any time, t, is Gaussian that is only characterized by its mean vector and its covariance matrix. Thus Eq. (2.3) and Eq. (2.4) could be written as,

$$X_t = F_t\, X_{t-1} + v_{t-1} \tag{2.5}$$

$$Z_t = H_t\, X_t + w_t \tag{2.6}$$

where F_t and H_t are known linear matrices that define the linear functions. The covariances of v_{t-1} and w_t are Q_{t-1} and R_t, respectively. Based on Eq. (2.5) and Eq. (2.6), the Kalman filter provides a straightforward and efficient recursion for computing the mean and covariance of the Gaussian *a posteriori* distribution of the object's state at the current time t given the measurement at t. In addition, it also provides the mean and covariance of the

a posteriori distribution of the object's state at $t-1$ as shown in the following recursive relationship

$$p(X_{t-1}|Z_{1:t-1}) = \text{Normal}\left(X_{t-1}; m_{t-1|t-1}, P_{k-1|k-1}\right) \tag{2.7}$$

$$p(X_t|Z_{1:t-1}) = \text{Normal}\left(X_t; m_{t|t-1}, P_{t|t-1}\right) \tag{2.8}$$

$$p(X_t|Z_{1:t}) = \text{Normal}\left(X_t; m_{t|t}, P_{t|t}\right) \tag{2.9}$$

where,

$$m_{t|t-1} = F_t\, m_{t-1|\mathrm{t}-1} \tag{2.10}$$

$$P_{t|t-1} = Q_{t-1} + F_t\, P_{t-1|t-1} F_t^T \tag{2.11}$$

$$m_{t|t} = m_{t|t-1} + K_t\left(z_t - H_t m_{t|t-1}\right) \tag{2.12}$$

$$P_{t|t} = P_{t|t-1} - K_t\, H_t P_{t|t-1} \tag{2.13}$$

$$K_t = P_{t|t-1} H_t^T \left(H_t P_{t|t-1} H_t^T + R_t\right)^{-1} \tag{2.14}$$

where $\text{Normal}(X; m, P)$ refers to a multivariate Gaussian density with argument X, mean vector m, and covariance matrix P; $m_{t|t-1}$ and $P_{t|t-1}$ are the mean and covariance of Gaussian *a posteriori* distribution at the current time *t* given all measurements up to the previous time $t-1$; and K_t is called the Kalman gain. If the highly-restrictive assumptions of Gaussian statistics and *F* and *H* being linear functions hold, the Kalman filter will be the optimal solution to the sequential Bayesian state estimation problem [17].

2.3.1.2 Grid-based methods

Grid-based methods give optimal solutions to the sequential Bayesian state estimation problem if the state space is discrete and comprises a finite number of states. Assuming the state space is composed of discrete states X_{t-1}^{i} at time *t - 1*, where $i \in \{1, \dots, M\}$ and M is the total number of states, then the conditional probability at state X_{t-1}^{i}, given measurements up to $t-1$, is $p\left(X_{t-1} = X_{t-1}^{i} \middle| Z_{1:t-1}\right) = w_{t-1|t-1}^{i}$. Therefore the *a posteriori* distribution at $t-1$ can be obtained by,

$$p(X_{t-1}|Z_{1:t-1}) = \sum_{i=1}^{M} w_{t-1|t-1}^{i} \ \delta\left(X_{t-1} - X_{t-1}^{i}\right) \tag{2.15}$$

where $\delta(.)$ refers to the unit impulse function. By substituting Eq. (2.15) into Eq. (2.3) and Eq. (2.4), the prediction and update equations can be written as,

$$p(X_t|Z_{1:t-1}) = \sum_{i=1}^{M} w_{t|t-1}^{i} \ \delta\left(X_t - X_t^{i}\right) \tag{2.16}$$

$$p(X_t|Z_{1:t}) = \sum_{i=1}^{M} w_{t|t}^{i} \ \delta\left(X_t - X_t^{i}\right) \tag{2.17}$$

where,

$$w_{t|t-1}^{i} = \sum_{j=1}^{M} w_{t-1|t-1}^{j} \ p\left(X_t^{i} \middle| X_{t-1}^{j}\right) \tag{2.18}$$

$$w_{t|t}^{i} = \frac{w_{t|t-1}^{i} \, p\left(Z_t \middle| X_t^{i}\right)}{\sum_{j=1}^{N} w_{t|t-1}^{j} \, p\left(Z_t \middle| X_t^{j}\right)} \tag{2.19}$$

We note that Grid-based methods do not require either $p(X_t^i|X_{t-1}^j)$ or $p(Z_t|X_t^i)$ to have a particular distribution. They will provide an optimal solution to the sequential Bayesian state estimation problem, if the state space is discrete and comprises a finite number of states.

2.3.2 Suboptimal filters

In many problems, the above assumptions are unrealistic and difficult to hold, so suboptimal filters can be used to approximate the integrals in Eq. (2.3) and Eq. (2.4). These suboptimal Bayesian filters include:

1. Extended Kalman Filter.
2. Approximate Grid-based methods.
3. Particle filters.

2.3.2.1 Extended Kalman Filter

In the case of nonlinear H or F functions, i.e., f or h, the Extended Kalaman Filter (EKF) replaces these functions with their linear approximations. For example, local linearization can be used to approximate these functions, then a Kalman filter can be used to find an approximate *a posteriori* Gaussian distribution as follows

$$p(X_{t-1}|Z_{1:t-1}) \approx \text{Normal}(X_{t-1}; m_{t-1|t-1}, P_{k-1|k-1}) \tag{2.20}$$

$$p(X_t|Z_{1:t-1}) \approx \text{Normal}(X_t; m_{t|t-1}, P_{t|t-1}) \tag{2.21}$$

$$p(X_t|Z_{1:t}) \approx \text{Normal}(X_t; m_{t|t}, P_{t|t}) \tag{2.22}$$

where,

$$m_{t|t-1} = f_t\ (m_{t-1|t-1})$$ (2.23)

$$P_{t|t-1} = Q_{t-1} + \hat{F}_t\ P_{t-1|t-1}\hat{F}_t^T$$ (2.24)

$$m_{t|t} = m_{t|t-1} + K_t(z_t - h_t(m_{t|t-1}))$$ (2.25)

$$P_{t|t} = P_{t|t-1} - K_t\ \hat{H}_t\ P_{t|t-1}$$ (2.26)

$$\hat{F}_t\ = \left.\frac{df_t(X)}{dX}\right|_{X=\,m_{t-1|t-1}}$$ (2.27)

$$\hat{H}_t\ = \left.\frac{dh_t(X)}{dX}\right|_{X=\,m_{t|t-1}}$$ (2.28)

where $\hat{F}_t$ and $\hat{H}_t$ are the local linearization of the non-linear functions, $f_t(.)$ and $h_t(.)$. These approximate linearizations use the first term only of the Taylor expansions of $f_t(.)$ and $h_t(.)$. A higher-order EKF could use higher terms of these Taylor expansions. This higher order EKF could lead to more accurate linearizations, but its additional complexity could be an obstacle to practical implementation.

2.3.2.2 Approximate Grid-based methods

In the case of a continuous state space that can be decomposed into *M* cells, i.e., a state space consisting of discrete states X_t^i and $i\ \in \{1, \dots, M\}$, an approximate Grid-based method can be used to approximate the *a posteriori* distribution. Assuming that an approximation of the *a posteriori* distribution at the previous time $t-1$ is given by

$$p(X_{t-1}|Z_{1:t-1}) \approx \sum_{i=1}^{M} w_{t-1|t-1}^{i}\ \delta\left(X_{t-1} - X_{t-1}^{i}\right). \tag{2.29}$$

Eq. (2.29) can be substituted into Eq. (2.3) and Eq. (2.4) to obtain the following prediction and update equations

$$p(X_t|Z_{1:t-1}) \approx \sum_{i=1}^{M} w_{t|t-1}^{i}\ \delta\left(X_t - X_t^{i}\right) \tag{2.30}$$

$$p(X_t|Z_{1:t}) \approx \sum_{i=1}^{M} w_{t|t}^{i}\ \delta\left(X_t - X_t^{i}\right) \tag{2.31}$$

where,

$$w_{t|t-1}^{i} = \sum_{j=1}^{M} w_{t-1|t-1}^{j}\ \int_{X \in x_t^i} p\left(X|\bar{X}_{t-1}^{j}\right) \tag{2.32}$$

$$w_{t|t}^{i} = \frac{w_{t|t-1}^{i}\ \int_{X \in X_t^i} p(Z_t|X)\, dX}{\sum_{j=1}^{N} w_{t|t-1}^{j}\ \int_{X \in X_t^j} p(Z_t|X)\, dX} \tag{2.33}$$

where $\bar{X}_{t-1}^{j}$ is the center of the j^{th} cell at time $t-1$. Since grid cells represent regions of a continuous state space, the probabilities in Eq. (2.32) and Eq. (2.33) must be integrated over these grid cells. The weight of the cell X_t^i is computed at its center, $\bar{X}_{t-1}^{i}$, as a further approximation in order to simplify the computation of weights:

$$w_{t|t-1}^{i} \approx \sum_{j=1}^{N} w_{t-1|t-1}^{j}\ p\left(\bar{X}_t^i|\bar{X}_{t-1}^{j}\right) \tag{2.34}$$

$$w_{t|t}^{i} \approx \frac{w_{t|t-1}^{i}\ \int_{X \in X_t^i} p\left(Z_t|\bar{X}_t^i\right) dX}{\sum_{j=1}^{N} w_{t|t-1}^{j}\ \int_{X \in X_t^j} p\left(Z_t|\bar{X}_t^j\right) dX} \tag{2.35}$$

However, approximate grid-based methods have some limitations. The grid must have a sufficiently large number of cells to produce a good approximation of the continuous state space, but the computational cost can increase dramatically with the dimension of the state space. Also, the state space grid must be *a priori* defined, so it cannot be dynamically redefined to obtain a higher resolution partitioning of the state space.

2.3.2.3 Particle filter

A particle filter approximates the *a posteriori* distribution, $p(X_t|Z_{1:t})$, using a point mass function representation, i.e., a weighted sum of samples called particles, $\{X_t^{(i)}\}_{i=1}^{N_s}$ associated with weights $\{w_t^{(i)}\}_{i=1}^{N_s}$. These samples are obtained using a simple update of samples at time t – 1, and these weights are calculated using the principle of *importance sampling* [18]. The particle filter then obtains the *a posteriori* mean from this approximate distribution using the Law of Large Numbers [19, 20]. This type of filter has many names; for example, it is also known as a *sequential Monte Carlo filter*, a *bootstrap filter*, or *the condensation algorithm*.

A particle filter boasts a simple parallelizable implementation that is independent of both the dimensions and linearity of Eq. (2.3) and Eq. (2.4). Also, it can be viewed as information fuser due to its ability to combine observational data and a dynamic model for a given object into one mathematical framework. Moreover, this ability can be extended to provide a general framework for data fusion.

2.3.2.3.1 Mont Carlo sampling

The Monte Carlo (MC) method is considered the basis for particle filters. The MC method is a numerical integration method that uses finite summations to estimate difficult definite integrals. Consider the function $g(X)$, which is dependent on a random variable X. Then its expected value,

$$E[g(X)] = \int g(\mathrm{X})\, p(\mathrm{X})\, dX \tag{2.36}$$

where $p(\mathrm{X})\, dX$ is the probability of the random variable X, has a value within dX of about X. If $g(X)$ is a complicated function, then it will be difficult to evaluate the expected value, $E[g(X)]$. The MC method can be used to approximate $E[g(X)]$ by a set of *independent and identically distributed* (IID) samples also known as particles. The MC method estimates $g(X)$ as

$$\hat{g} = \frac{1}{M}\sum_{m=1}^{M} g(X^m) \tag{2.37}$$

where $X^m \sim p(.)$, and M is the total number of particles. Consequently, the expected value of the MC estimate is

$$E[\hat{g}] = E\left[\frac{1}{M}\sum_{m=1}^{M} g(X^m)\right]$$

$$= \frac{1}{M} E\left[\sum_{m=1}^{M} g(X^m)\right]$$

$$= \frac{1}{M} \sum_{m=1}^{M} E[g(X^m)]$$

$$= E[g(X)] \tag{2.38}$$

Based on the strong law of large numbers [19, 20], the MC estimate converges to the true value of $E[g(X)]$ as $M \to \infty$. However, the MC method cannot be used alone in sequential Bayesian estimation problems, as the distribution $p(.)$ from which samples are drawn is unknown. This problem is handled via importance sampling, which will be introduced in the next section.

2.3.2.3.2 Importance sampling

When it is difficult to draw directly samples from a *target distribution* $p(\cdot)$, *importance sampling* (IS) aims to represent a $p(\cdot)$ by drawing weighted particles $\{X_t^{(i)}\}_{i=1}^{N_s}$ associated with weights $\{w_t^{(i)}\}_{i=1}^{N_s}$. This is achieved by drawing samples from a *proposal distribution* $q(\cdot)$, also called an *importance distribution*, where $p(\cdot) \propto q(\cdot)$. Consider the following integral

$$\int g(x)p(x)\,dx = \int g(x)\,\frac{p(x)}{q(x)}q(x)\;\,dx \tag{2.39}$$

Based on the MC method, $p(x)$ is represented as,

$$p(x) \approx \sum_{m=1}^{M} w^m\,\delta(X - X^m) \tag{2.40}$$

$$w^m = \frac{p(X^m)}{q(X^m)} \tag{2.41}$$

where w is the importance weights. The MC estimate of $g(X)$ is shown in Eq. (2.42), where the weight is normalized to ensure that $\sum_{m=1}^{M} \widetilde{w}(X^m) = 1$.

$$\hat{g} = \frac{1}{M} \sum_{m=1}^{M} \widetilde{w}(X^m)\, g(X^m) \tag{2.42}$$

$$\widetilde{w}(X^m) = \frac{w(X^m)}{\sum_{m=1}^{M} w(X^m)} \tag{2.43}$$

For Bayesian filtering, the *a posteriori distribution* $p(x_t|z_{1:t})$ can be represented by a weighted set of particles $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$, which are drawn from a known proposal distribution $q(x_t|z_{1:t-1})$. The weights of this set of particles are calculated as in Eq. (2.41). Hence, the *a posteriori* distribution at t can be approximated based on Eq. (2.44). A *sequential importance sampling* (SIS) is then used to recursively propagate the weighted particles. A description of an SIS is introduced in the next section.

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)}\, \delta\left(x_t - x_t^{(i)}\right) \tag{2.44}$$

$$w_t^{(i)} \propto \frac{p\left(x_t^{(i)}|z_{1:t}\right)}{q\left(x_t^{(i)}|z_{1:t}\right)} \qquad i = 1, \ldots, N_s \tag{2.45}$$

2.3.2.3.3 Sequential importance sampling particle filter

The aim of SIS is to find the weighted set of particles $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$ at the current time t that represents the *a posteriori* $p(x_t|z_{1:t})$ given weighted particles $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_s}$

at the previous time $t-1$, which in turn represent the previous *a posteriori* $p(x_{t-1}|z_{1:t-1})$. To find the current *a posteriori* distribution, a prediction and an update step are required. The prediction step propagates the particle $\{x_t^{(i)}\}_{i=1}^{N_s}$ from time $t-1$ to time t by selecting the factorized *importance density* via

$$q(X_{0:t}|Z_{1:t}) = q(X_t|X_{0:t-1}, Z_{1:t})\, q(X_{0:t-1}|Z_{1:t-1}) \tag{2.46}$$

Eq. (2.46) implies that the samples $x_{0:t}^{(i)} \sim q(X_{0:t}|Z_{1:t})$ are obtained by augmenting the previous samples $x_{0:t-1}^{(i)} \sim q(X_{0:t}|Z_{1:t-1})$ with the new state $x_t^{(i)} \sim q(X_t|X_{0:t-1}, Z_{1:t})$. The update step computes the weights $\{w_t^{(i)}\}_{i=1}^{N_s}$ using the newly-introduced measurement Z_t. The *a posteriori* distribution is represented as

$$\begin{aligned}
p(X_{0:t}|Z_{1:t}) &= \frac{p(Z_{1:t}|X_{0:t})\, p(X_{0:t})}{p(Z_{1:t})} \\
&= \frac{p(Z_t|X_{0:t}, Z_{1:t-1})\, p(Z_{1:t-1}|X_{1:t}) p(X_{0:t})}{p(Z_t|Z_{1:t-1}) p(Z_{1:t-1})} \\
&= \frac{p(Z_t|X_{0:t}, Z_{1:t-1})\, p(X_{0:t}|Z_{1:t-1})\, p(Z_{1:t-1})\, p(X_{0:t})}{p(Z_t|Z_{1:t-1})\, p(Z_{1:t-1})\, p(X_{0:t})} \\
&= \frac{p(Z_t|X_t)\, p(X_{0:t}|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \\
&= \frac{p(Z_t|X_t)\, p(X_t|X_{0:t-1}, Z_{1:t-1})\, p(X_{0:t-1}|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \\
&= \frac{p(Z_t|X_t)\, p(X_t|X_{0:t-1})\, p(X_{0:t-1}|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \\
&\propto p(Z_t|X_t)\, p(X_t|X_{0:t-1})\, p(X_{0:t-1}|Z_{1:t-1})
\end{aligned} \tag{2.47}$$

By substituting Eq. (2.46) and Eq. (2.47) into Eq. (2.45), the updated weight can be represented as,

$$w_t^{(i)} \propto \frac{p\left(Z_t \middle| X_t^{(i)}\right) p\left(X_t^{(i)} \middle| X_{0:t-1}^{(i)}\right) p\left(X_{0:t-1}^{(i)} \middle| Z_{1:t-1}\right)}{q\left(X_t^{(i)} \middle| X_{0:t-1}^{(i)}, Z_{1:t}\right) q\left(X_{0:t-1}^{(i)} \middle| Z_{1:t-1}\right)}$$

$$= \widetilde{w}_{t-1}^{(i)} \frac{p\left(Z_t \middle| X_t^{(i)}\right) p\left(X_t^{(i)} \middle| X_{0:t-1}^{(i)}\right)}{q\left(X_t^{(i)} \middle| X_{0:t-1}^{(i)}, Z_{1:t}\right)} \qquad i = 1, \dots, N_s \tag{2.48}$$

where $\widetilde{w}_{t-1}^{(i)}$ is the normalized weight, *i.e.*, $\widetilde{w}_{t-1}^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N_s} w_t^{(i)}}$. The *a posteriori* $p(x_t|z_{1:t})$ is then approximated by

$$p(x_t|z_{1:t}) = p(x_t|z_{1:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \, \delta\left(x_t - x_t^{(i)}\right) \tag{2.49}$$

2.3.2.3.4 Sequential importance resampling particle filter

Unfortunately, after a few iterations of running a sequential importance sampling particle filter, all but one particle will have almost zero weight. This is called Degeneracy problem [17]. Consequently, a high computational effort is required to update particles that have negligible weights and that do not contribute to the correct approximation of the *a posteriori* distribution. Resampling is one method of mitigating this effect, and it does so by eliminating particles that have small weights and concentrating on those with large weights.

Table 2.1 shows the complete sequential importance resampling (SIR) particle filter in which the importance distribution, $q(X_t|X_{0:t-1}, Z_{1:t})$, is the prior distribution, $p(X_t|X_{t-1}^{(i)})$ [14].

Table 2.1. Standard particle filter using sequential importance resampling

Step 1: Initialization step, t=0

- For $i = 1, \ldots, N_s$ sample $X_t^{(i)} \sim p(X_0)$
- Set $t = 1$

Step 2: Propagation step

- For $i = 1, \ldots, N_s$ sample $x_t^{(i)} \sim p(X_t|X_{t-1}^{(i)})$

Step 3: Importance sampling step

- For $i = 1, \ldots, N_s$ compute the importance weights $w_t^{(i)} = p(Z_t|\, X_t^{(i)})$
- Normalize the importance weights $\sum_i \widetilde{w}_t^{(i)} = 1$

Step 4: Resampling step

- According to the normalized weights, resample with replacement N_s particles $\{X_t^{(i)}\}_{i=1}^{N_s}$ from $\{\tilde{X}_t^{(i)}\}_{i=1}^{N_s}$.
- Set $t = t + 1$ and go to Step 2

2.3.2.4 Limitations of the particle filter for robust visual tracking

While the standard particle filter is a typical method used for visual tracking, it has many limitations when tracking multiple objects in the presence of the challenging conditions and unexpected events mentioned in Section 1.1. For example:

1. the presence of background motion or object shadows can lead to additional

(spurious) likelihood modes,

2. the presence of objects with different sizes or contrast levels can lead to a dominant likelihood problem, where a posterior probability distribution will be dominated by a single object (likelihood mode) with the largest size or highest brightness,
3. the presence of high noise levels in video frames can create additional (spurious) likelihood modes,
4. sudden changes in illumination can cause sudden changes in the likelihood function, and
5. the presence of partial object camouflage can lead to sudden changes in the likelihood modes.

In the next chapters, we will introduce our novel approaches that can be combined with the standard particle filter to overcome these limitations.

2.3.3 Literature review for Bayesian visual tracking

A variety of different visual trackers that are based on the Bayesian approach have been proposed. This section represents a general literature review of the existing visual trackers using a Bayesian approach. We will emphasize particle filter-based visual trackers because they can handle nonlinear models and non-Gaussian models, and can focus on higher-density regions of the state space as discussed in Section 2.3.2.3. Also, these particle filter-based trackers are parallelizable and easy to implement. Furthermore, particle filter-based trackers typically provide better tracking performance than other Bayesian Bayesian trackers using Kalman Filters, EKFs, and grid-based methods.

Kalman filter based visual trackers

The authors in [21] used an Extended Kalman Filter to estimate the motion model parameters of a rigid body in a sequence of noisy images. They modeled the dynamics of an object as a nonlinear function of time before estimating the parameters of this motion model. We note that the motion model described two types of object motion, including 3-D rotation and translational motion.

In [22], the authors introduced a visual tracker that is capable of categorizing different types of object motion. They used a 3-D linear trajectory model in conjunction with an Extended Kalman Filter to estimate the 3-D motion trajectories for objects based on their 2-D motion. Using the obtained 3-D trajectory, they were able to construct stabilized views of the moving object, which allowed them to recognize various dynamic human activities, such as running, walking, roller skating, and cycling.

The authors in [23] described a visual tracker based on the Hough transform [24] and an Extended Kalman Filter. The Hough transform could be used as a feature extraction technique that identifies imperfect instances of objects, usually parametrized in polar coordinates, within particular classes of shapes, e.g., ellipses, circles, and lines. The authors used an EKF to model the parameters and motion of a set of lines detected in Hough space. The use of the Hough transform increased the tracker's resilience to noise and partial occlusion, and the EKF reduced the computational load required for line detection.

The work in [25] presented a visual motion tracker for 3-D objects that overcomes the uncertainty of varying noise statistics due to the poor quality of the camera sensor. The authors used an Adaptive Extended Kalman Filter after adding more variables to be

estimated by the filter, e.g., noise statistics. These additional variables allowed real-time adaptation to the time-varying statistics of both state and observation noise.

The authors in [26] introduced a stereo camera-based visual tracker, wherein real-time stereo information could be used for 3-D continuous detection and tracking, even in a cluttered scene. They applied a Kalman filter for predicting and updating the object's position in the current frame. After using a stereo-processing unit to perform the area-based correlation, objects were then detected using a background subtraction technique.

The authors in [27] proposed a corner feature-based visual tracker using an Adaptive Kalman Filter. This tracker represented a moving object using a set of corners. It then used the variation in the number of occluded corner points across consecutive frames to automatically adjust the parameters of the Kalman filter.

The work in [28] presented a two-stage visual tracker that combined template-matching and contour methods. In the first stage, a Kalman filter predicted the initial object position. Then it used a template-matching method where a color histogram model was extracted from a candidate region centered at the position predicted by the Kalman filter. A *mean shift* algorithm then found the object's new position by maximizing the Bhattacharyya coefficient between the extracted color histogram and color histograms of different frame regions. In the second stage, the active contour representing the object was evolved to improve tracking precision.

The authors in [29] introduced a visual tracker based on Kalman filtering, online feature selection, and a *mean shift* algorithm. An object was defined by its positions in x and y, scale, and orientation where changes these characteristics were tracked. First, a

Kalman filter estimated the object's state. Then using color pixel values in the R, G, and B frames 28 features were then constructed, followed by online feature selection that best-distinguished objects and background scenes. Finally, the *mean shift* algorithm was used to find the new location of the object.

The work in [30] described a 3-D visual tracker that used a sequence of depth images obtained by a Microsoft Kinect camera. Two visual trackers were developed based on motion image segmentation, and a *mean shift* tracking algorithm, respectively. A Kalman filter was then used to fuse the two resulting motion tracks.

The authors in [31] developed a visual tracker for use in Micro Aerial Vehicles (MAVs). This tracker used an Extended Kalman Filter to estimate the state, i.e., position and velocity, of moving objects on the ground. This EKF initialized the state ground objects using an object detection algorithm. The EKF then predicted the object's state and used a nonlinear measurement model to update it.

Grid-based visual trackers

The authors in [32] presented a visual tracker that utilized *Channel-Based Bayesian* Tracking, which is a generalization of grid-based methods [33]. This approach reduces computation cost of the grid-based method by using grids with fewer cells. The used samples were overlapped due to sampling with smoothed kernel functions instead of impulse functions.

The work in [34] presented a visual tracker that used a grid-based method to track leukocytes *in vivo* and to observe vehicles from an *Unmanned Aerial Vehicle* (UAV).

Assuming smooth objects' motion trajectories, it generated a deterministic set of samples in ellipsoidal areas centered at the object positions predicted by a *motion model*. These samples were then weighted by their distances from the predicted position of the object, and visual features, e.g., object boundary, were detected using a *radial edge detector*. This tracker was able to handle background movement, image clutter, and occlusion. However, the movement of an object cannot be erratic, i.e., its velocity cannot change abruptly, because of the smoothness constraint on its trajectory. To handle the possibility of erratic motion of the objects described in [34], another grid-based visual tracker was developed in [35]. If an object moves erratically, it will tend to escape the coverage of its ellipsoidal grid area. Therefore, if the difference between the predicted and updated positions is large, an erratic motion is detected, and the *motion model* will be modified accordingly.

Particle filter-based visual trackers

Many visual trackers using different features, e.g., shape and/or color, and different versions of the particle filter, e.g., Interacting Multiple Model (IMM) particle filter, Auxiliary particle filter, or Markov Chain Monte Carlo (MCMC) particle filter.

The authors in [36] introduced a multiple-object visual tracker based on a particle filter that could handle the presence of occlusion, clutter background, and changes in an object's appearance by considering the objects' predicted trajectories using a dynamic model and likelihood functions. The likelihood function was based on a color histogram model, where a measure of similarity was evaluated based on the Bhattacharya coefficient between a reference color histogram model for the object and a color histogram model from a candidate object region. We note that the authors updated the reference object model in

each frame, and concluded the presence of occlusion if the distance between any pair of tracked objects was less than a specific value. In the case of object occlusion, object positions were updated based on the motion model only.

The authors in [37] developed a 2-D visual tracker for articulated objects, e.g., a human body, using a particle filter. They used a partition-sampling method to divide a high-dimensional space into two or more subspaces, before tracking the object in each sub-space. Also, the authors developed a new Belief Propagation (BP) method that enabled a set of particles to fulfill several constraints on, e.g., the distance between neck and shoulder.

The work in [38] proposed a real-time hybrid visual tracker that could cope with the presence of objects in a crowded scene. It integrated a *blob*-based tracker and a color-based particle filter tracker. The *blob*-based tracker was the main component, while the color-based particle filter tracker was only invoked in cases of object merging and/or occlusion. In the blob tracker, a motion map of the objects was extracted using a background subtraction method. Once this motion map has been extracted, the current map was compared to the previous map to determine one of four events, i.e., a new object, an existing object, blob splitting, or blob merging. If a merging blob event were detected, the tracker would invoke the particle filter tracker. The likelihood function used by this particle filter evaluated a similarity score based on the Bhattacharya coefficient between a reference color histogram model, obtained from the previous frame, and a color histogram model from a candidate object region. This particle filter assumed that occluded objects had the same color appearance model as before their occlusion.

The authors in [39] introduced a visual tracker to track humans in 3D. They used a progressive particle filter that used hierarchical searching to decrease the computational costs for human body configurations with high degrees of freedom. They used a likelihood function based on silhouette masking, edge distance mapping, contour distance mapping, and skin color mapping. Also, they used a *mean shift* algorithm to increase state estimation accuracy by moving each particle toward the location with the highest probability of posture.

The work in [40] introduced a visual tracker based on a modified particle filter for tracking pedestrians using infrared video sequences. It constructed two likelihood functions, one based on an intensity histogram model, and another based on an edge model; then it combined them using an adaptive weighted sum strategy.

The authors in [41] introduced a visual tracker, where they implemented an efficient particle filter's resampling step using Particle Swarm Optimization (PSO) [42]. They used the PSO algorithm to explore the area around the object's previous position, where, to achieve diversity and convergence, particles were distributed using two different base points. We note that the used likelihood function was based on a color histogram model and a histogram of the gradients of the object's orientations.

The work in [43] described a visual tracker that used a particle filter that sampled particles from the object's posterior distribution using an MCMC sampling method. This avoided the sample impoverishment problem and enhanced the robustness of the particle filter. In this tracker, each particle was propagated based on both its history in addition to information from other particles.

The authors in [44] described a particle filter based visual tracker that used adaptive integration of both object color histogram, and object contour information obtained using a Sobel operator, in the likelihood function. This likelihood function computed the similarity between an object's color histogram and contour information and candidate object positions using a Bhattacharyya distance.

The work in [45] presented a visual tracker that used an object color histogram and Harris corners to obtain an object's rough location. Then it extracted the object's contours with the help of a region-based object contour extraction algorithm. Next, this tracker used object contours with a particle filter to obtain the object's accurate location based on this rough location.

The authors in [46] developed a visual tracker based on object *Scale Invariant Feature Transform* (SIFT), and object color histogram features. SIFT features provide a key feature that is invariant to image scaling, translation, and rotation, and can be used for object matching and localization. First, the user selected an object region by drawing a rectangle around it, where SIFT features were extracted for object representation and localization in the next frame. Second, this tracker applied a particle filter using a color histogram model to estimate object positions in the following frame.

In [47], the authors introduced a visual tracker that built a reference model for an object in a manually selected region. They then divided this chosen region into N sub-regions where they extracted SIFT features from all of them. These SIFT features were used to construct a multi-part SIFT reference model for this object. Afterward, they applied a particle filter that used this SIFT-based likelihood function. In every frame, this SIFT-

based likelihood function was updated by replacing its object reference model by the object model from the previous frame with the highest matching score.

The authors in [48] compared different trackers with particle filters using color histogram based likelihood functions. They considered several color spaces, including RGB, HSV, and YCbCr. Their comparison demonstrated that the HSV color space is optimal for tracking objects with scale variation, occlusion, in backgrounds exhibiting illumination change.

The work in [49] described a visual tracker to track a selected object in an environment with multiple moving objects. To detect the presence of moving objects, an initial background extraction through averaging video frames for a particular time period and background subtraction was used. After object detection, only a single object was selected to be tracked. They used a particle filter to estimate the selected object's position. The likelihood function computes the similarity between a reference window around the selected object and a window around a candidate object region. We note that the background was updated in every frame to ensure accurate detection of moving objects.

The authors in [50] described a particle filter based visual tracker that represented objects based on regions that were homogeneous in color. The object tracking algorithm segmented its shape in all frames of the video sequence. This image segmentation allowed for updating the object model, and for dealing with change in both color and shape of the object.

The author of [51] described a visual tracker that used a *mean shift* algorithm to construct the likelihood function of the particle filter. This *mean shift* algorithm maximized

the Bhattacharyya distance between the color histograms of regions where an object is located in the current and previous frames [52]. The performance of this tracker, however, degraded significantly during long periods of partial object occlusion.

The visual tracker discussed in [53] used a color-based object appearance model. The coefficients of this color model were included in the state vector of the object (augmented state vector) and were updated adaptively, along with the original state vector, upon the arrival of a new frame. To avoid the potential curse of dimensionality due to the augmentation of the object's state vector, a Rao-Blackwellized particle filter [20] was used to estimate both the state and coefficients of the object's color model. A Rao-Blackwellized particle filter allows the color model's coefficients to be computed analytically and the object's state to be estimated numerically. This tracker, however, required intensive computational effort.

To cope with an abrupt change of illumination and the presence background motion, the work in [54] fused different object features into a single likelihood function. These features consisted of color histogram distribution, edge information, and structure information. The structure information was built using a boosted multi-view shape detector [55] for particular types of objects, such as human faces. Therefore, to track a different kind object, such as cars, the boosted multi-view shape detector required another offline training.

The work documented in [56] was oriented towards handling changes in illumination. To this end, the authors designed a visual tracker that fused an object's appearance and shape models using a particle filter. At each time instance, the tracker described the

object's appearance using two features, i.e., the color histogram distribution, and the object's contours. It then applied a particle filter to process both features sequentially, re-used the estimate, estimated the posterior, and then produced one feature as initial input for the next feature processing step. The final object position track was the product of the estimated posterior densities.

Another robust visual tracker based on the fusion of multiple object features was developed in [57] to address the presence of objects with different sizes or contrast levels and the presence of partial object camouflage. The likelihood function of this tracker's Bayesian model represented human motion using multiple object features, including color and edge shape. Because the models for these object features were generated off-line before the tracking started, this tracker was limited to tracking a single type of object, for instance, a human body.

Another work in [58] introduced a real-time robust visual tracker based on the fusion of data from multiple sensors. This visual tracker demonstrated robustness in tracking randomly moving objects in real-time. The sensors included a pan-tilt camera and sixteen sonar sensors. However, it failed to track fast-moving objects as they passed out of the camera's field of view.

2.4 Chapter summary

This chapter described three main approaches to visual tracking: point tracking, kernel-based tracking, and silhouette-based tracking. It mainly focused on Bayesian statistical approaches to point tracking. An extensive review of the literature on Bayesian visual trackers was presented. As discussed, particle filter based visual trackers can handle

nonlinear motion and/or nonlinear measurement models, and/or non-Gaussian statistics. Compared to other statistical approaches, e.g., Kalman Filters, EKFs, and grid-based methods, particle filter based visual trackers are easier to implement and offer better tracking performance.

Chapter 3

Robust Visual Tracking

3.1 Introduction

Over the past few decades, there have been remarkable advances in visual tracking. Nonetheless, robust visual tracking remains an active research topic [3, 4]. Robust visual tracking refers to a tracking device's ability to avoid tracking failures [59], and to track objects accurately in video sequences that have unexpected events and challenging conditions [2, 10]. These difficult circumstances could include 1) the presence of background motion and shadows; 2) the presence of objects with different sizes and contrast levels; 3) the presence of partial object camouflage; 4) sudden changes in scene illumination; 5) low signal-to-noise (SNR) ratios; and 6) real-time processing requirements.

To address these conditions, we discuss three approaches to robust visual tracking: a multi-scale approach using wavelets, a fusion based approach as well as machine learning based approach.

In this chapter, Section 3.2 discusses and reviews the literature on multi-scale approach to robust visual tracking in the wavelet domain. Section 3.3 discusses fusion approach for robust visual tracking. Section 3.4 discusses machine learning approach for robust visual tracking. Finally, we summarize the contents of this chapter in Section 3.5.

3.2 Multi-scale approach using wavelet domain for robust visual tracking

The wavelet transform is important for multi-scale analysis, also known as multi-resolution analysis (MRA), of signals or images [60, 61]. The wavelet transform represents an image using different resolution levels. Therefore, different features in a full-resolution image can be analyzed at different image scales [61]. The discrete wavelet transform was used in many applications, such as image denoising [62], image compression [63], and image texture classification [64].

In the following sub-sections, the discrete wavelet transform is introduced, and the advantages of multiple-object tracking in the wavelet domain are discussed.

3.2.1 Discrete wavelet transform

The wavelet transform represents a signal, $f(x)$, in the space-frequency domain as a weighted sum of space-frequency atoms with different scales [65, 66]. To perform a wavelet transform, a *scaling function*, $\emptyset(x)$, is used to generate a series of approximations of the original signal. An additional function, known as a *wavelet function*, $\psi(x)$, is used to represent the difference between two successive approximations [61, 67].

3.2.2 Bases, frames and linear expansions of signals

A linear vector space V could be spanned by a set of vectors, $p_k(x)$, if any element or function, $f(x)$, in the space can be expressed as a linear of combination of this set [10] as shown in Eq. (3.1).

$$V = \left\{ f(x) = \sum_k \alpha_k p_k(x) \right\} \tag{3.1}$$

Where α_k are the coefficients of this expansion. The set, $\{p_k(x)\}$, is called a basis for a given space V if α_k are unique for the given $f(x) \in V$. The *closed span* of $\{p_k(x)\}$ refers to all possible functions $f(x) \in V$ that can be expressed by as a linear combination of $\{p_k(x)\}$, and can be written as

$$V = \overline{span\{p_k(x)\}}. \tag{3.2}$$

The computation of the expansion coefficients α_k depends on the expansion set. Typically, there are three cases for $\{p_k(x)\}$ [61, 68]:

Case 1: The expansion functions are orthonormal, i.e.,

$$\langle p_j(x), p_k(x) \rangle = \delta_{ik} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases} \tag{3.3}$$

where $\langle \cdot \rangle$ denotes inner product. Therefore, the expansion coefficients α_k are given by the inner products of $\{ p_k(x)\}$ and $f(x)$

$$\alpha_k = \langle p_j(x), f(x) \rangle \tag{3.4}$$

Case 2: The expansion functions are linearly independent but are not orthonormal, i.e.,

$$\langle p_j(x), p_k(x) \rangle \neq 0 \qquad j \neq k \tag{3.5}$$

The expansion coefficients, α_k, can be computed using a dual basis set, $\{\tilde{p}_k(x)\}$, whose functions are also linearly independent but not orthonormal, and satisfy the following condition

$$\langle p_j(x), \tilde{p}_k(x) \rangle = \delta_{ik} \tag{3.6}$$

In this case, the basis set and its dual basis are referred to as *biorthogonal* and the expansion coefficients, α_k , are computed as

$$\alpha_k = \langle f(x), \tilde{p}_k(x) \rangle \tag{3.7}$$

Case 3: The expansion functions and their duals are redundant, and comprise a *frame* that, for all $f(x) \in V$, satisfies

$$A\|f(x)\|^2 \leq \sum_k |\langle p_k(x), f(x) \rangle|^2 \leq B\|f(x)\|^2 \tag{3.8}$$

where $A > 0$ and $B < \infty$. If $A = B$, this expansion set is known as a *tight frame* and any function $f(x) \in V$ can be represented as

$$f(x) = \frac{1}{A} \sum_k \langle p_k(x), f(x) \rangle \; p_k(x) \tag{3.9}$$

3.2.3 Scaling functions

In this section, we will consider basis functions, with integer translation k and integer scaling factor j, called *scaling functions*

$$\emptyset_{j,k}(x) = 2^{\frac{j}{2}} \emptyset(2^j x - k) \tag{3.10}$$

The position of the scaling function $\emptyset_k(x)$ is determined by k, its width along the *x*-axis is determined by 2^j, and its amplitude is given by $2^{\frac{j}{2}}$. If j increases, $\emptyset_{j,k}(x)$ becomes narrower and is translated in smaller steps, and therefore it can represent finer signal details. On the other hand, if j decreases, $\emptyset_{j,k}(x)$ becomes wider and is translated in larger steps, and therefore it can represent coarser signal information [66, 69].

If j is set to a specific value, and the set $\{\emptyset_{j,k}(x)\}$ spans the space V_j, i.e.,

$$V_j = \overline{Span_k\{\emptyset_{j,k}(x)\}}, \tag{3.11}$$

then any given $f^j(x) \in V_j$ can be expressed as,

$$f^j(x) = 2^{\frac{j}{2}} \sum_k \alpha_k \emptyset(2^j x - k). \tag{3.12}$$

As per Mallat's observation that "*subspaces spanned by scaling function at low scales are nested within those spanned at higher scales*" [60], i.e., $V_{-\infty} \subset \cdots V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots V_{\infty}$, the basis functions of subspace V_j ,i.e., $\{\emptyset_{j,k}(x)\}$ can be expressed in terms of scaling functions of subspace V_{j+1}, i.e., $\{\emptyset_{j+1,n}(x)\}$, as

$$\emptyset_{j,k}(x) = \sum_n \alpha_n \emptyset_{j+1,n}(x) \tag{3.13}$$

By substituting $\emptyset_{j+1,n}(x)$ from Eq. (3.10) into Eq. (3.13) and changing the variable α_n to $h_\emptyset(n)$, we get the *refinement equation*,

$$\emptyset(x) = \sqrt{2} \sum_n h_\emptyset(n) \emptyset(2x - n) \tag{3.14}$$

where $h_\emptyset(n)$ are called the *scaling function coefficients*. The relationship between different nested spaces that are spanned by scaling functions is shown in Figure 3.1.

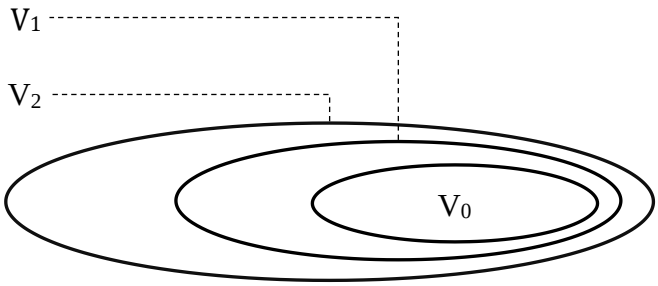


Figure 3.1. Nested spaces spanned by scaling functions

3.2.4 Wavelet functions

A set of wavelet functions, $\{\psi_{j,k}(x)\}$, defined by Eq. (3.15), can be used to span the space, W_j, which is the difference between two adjacent subspaces, V_j and V_{j+1}, i.e., $V_{j+1} = V_j \oplus W_j$, where $\oplus$ corresponds to space union [61].

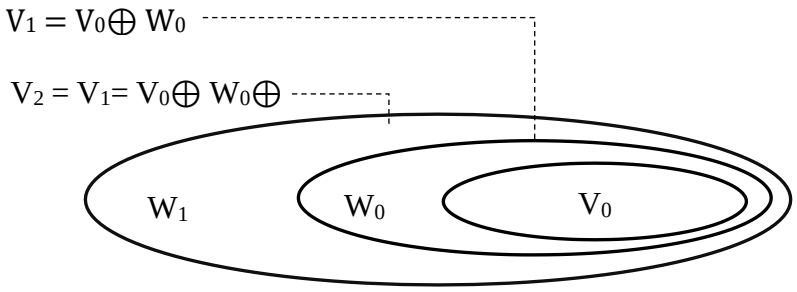


Figure 3.2. Difference between two adjacent subspaces is spanned by wavelet functions

It is clear from Fig. 3.2, that spaces W_j and V_j are orthogonal.

$$\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi(2^j x - k) \tag{3.15}$$

Similar to scaling functions, any wavelet function $\psi_{j,k}(x)$ can be expressed using a refinement equation.

$$\psi(x) = \sqrt{2}\sum_{n} h_{\psi}(n)\psi_{k}(2x-n) \tag{3.16}$$

where $h_{\psi}(n)$ are called the *wavelet function coefficients*.

Let $l^2(R)$ be the space of all square integrable functions,

$$l^2 = \mathrm{V}_{j_0} \oplus \mathrm{W}_{j_0} \oplus \mathrm{W}_{j_0+1} \oplus \ldots \tag{3.17}$$

then any function $f(x)$ in l^2 can be presented as

$$f(x) = \sum_{k} a_{j_0}(k)\,\emptyset_{j_0,k}(x) + \sum_{k}\sum_{j=j_0}^{\infty} d_j(k)\,\psi_{j,k}(x) \tag{3.18}$$

where j_0 is an arbitrary starting scale, and $a_{j_0}(k)$ and $d_j(k)$ are the coefficients of the DWT for the signal$f(x)$.

3.2.5 Advantages of multiple object tracking in the wavelet domain

Multiple object visual tracking in the wavelet domain has many advantages:

1. Video frames with coarse resolution (large scale) are more suitable for tracking large objects, and/or objects with high contrast, while video frames with fine resolution (small scale) are more suitable for tracking small objects and/or objects with low contrast [61]. Therefore, the wavelet transform of an original video frame produces subband frames with different resolutions (scales) that are suitable for tracking different types of objects that could be present in this same frame.
2. The wavelet transform is a natural edge detector that can detect boundaries of objects in various directions. The one-dimensional discrete wavelet transform

(DWT) could be easily extended to two dimensions, where the resulting separable functions, $\psi^H(x,y) = \psi(x)\,\emptyset(y)$, $\psi^V(x,y) = \emptyset(x)\psi(y)$ and $\psi^D(x,y) = \psi(x)\,\psi(y)$, would be sensitive to the horizontal, vertical, and diagonal edges, respectively [61, 70].

3. Wavelet denoising of images in the wavelet domain is simple and is typically performed by either hard or soft thresholding the DWT coefficients [71]. Instead of denoising white Gaussian noise only, a nonlinear multi-scale transform that combines the DWT with a median filter could be used to also reduce both speckle noise and salt and pepper noise [72].

3.2.6 Review of literature on robust tracking using the wavelet transform

The authors in [73] described a visual tracker that assumed a fixed object size. After selecting a region where the object occupied, they computed energies of the coefficients of a biorthogonal wavelet transform in this region. Then they compared them to candidate object regions in the next frame, where the region that gave the best match would become the new region for this object.

A robust tracker to cope with the presence of background motion and changes in illumination was described in [74]. A two-dimension Discrete Wavelet Transform (DWT) was applied to the given video sequence, but only the low-low subband of the third scale, i.e., $(\mathrm{LL})_3$ was used by this tracker. A visual tracker based on a similar idea, i.e., using lower resolution frames, to cope with the change of illumination and background motion was introduced in [75].

A low pass filter, 2×2 averaging filter, was applied to generate lower resolution frames by replacing each pixel value of the original image with the average value of itself and its neighbors. However, this averaging filter led to a more blurred frame compared to the $(\mathrm{LL})_3$ wavelet subband frame, thus decreasing tracking performance [75].

A similar visual tracker was also developed using the $(\mathrm{LL})_2$ wavelet subband, where a *direct LL-mask band scheme* (DLLBS) was used to speed up computation. The DLLBS scheme efficiently implements 2-D *symmetric mask-based discrete wavelet transform* (SMDWT) [76]. As effective tracking of objects with different contrasts and different sizes would require information from different DWT scales [61], robust tracking of such challenging objects would be unlikely using this method.

A multi-scale approach for detecting moving objects in a video was introduced in [77]. Optical flow was estimated at different scales via a quad-tree structure, where each node was represented as a linear combination of its parents. Even though this optical flow-based approach was effective in detecting moving objects in the video, it would be difficult and computationally intensive for real-time visual tracking [78].

In [79], the author proposed a rigorous formulation for visual tracking through multiple scales, developing a multi-scale tracker based on *dynamic Markov networks* and a pyramid decomposition of the video frames. The author performed a bi-directional propagation of the object's posteriors on different scales; that is, the tracking process in each scale interacted with its corresponding process in the previous and current time and its higher and lower scales at the same time. A color histogram distribution similar to the

one in [80] was used to model the likelihood function of the particle filter. However, the construction and maintenance of a *dynamic Markov network* require intensive computation that limited the use of this tracker to simple tracking scenarios.

The authors of [81] tried to reduce computation costs in comparison to the multi-scale tracker in [79]. They used sequential Bayesian filtering to propagate the object's posteriors across different resolution scales that were generated by pyramid decomposition.

3.3 Robust visual tracking using fusion

Fusion could be a possible solution for developing a more accurate and robust visual tracker [10, 82, 83] because it can be implemented using a fusion of

1. Different visual features from a video frame. This *visual feature fusion* uses multiple features from a video frame to represent an object to cope with possible changes in its appearance [10, 83].

2. Measurements from multiple sensors. This *data fusion* uses independent measurements from multiple sensors to develop a single visual tracker [1].

3. Different object motion models. This *motion model fusion* switches between several prior object motion models inside a single visual tracker via *Interactive Multiple Models* (IMM) [84].

4. Tracking paths resulting from several visual trackers. This *tracker fusion* involves running several (parallel or series) visual trackers, then combining their resulting tracking paths to obtain a final tracking path. These different trackers can operate

independently [16, 85, 86] or they can interact together [87].

Different strategies could be applied to perform any of the above types of fusion. For example, 1) a weighted sum rule [86, 88]; 2) a product rule [89, 90]. Using a weighted sum rule, the final outcome is obtained as a weighted average of outcomes due to the individual variables to be fused together. Using a product rule, dependencies between different variables to be fused are ignored, and their marginal probabilities are combined into a single joint probability distribution.

3.4 Robust visual tracking based on machine learning

Machine learning could be another possible approach to obtain robust visual trackers, where object tracking is formulated by learning different object views. Learning mechanisms aim to generate a function that maps inputs to desired outputs based on a set of features that will discriminate object classes. This set of features could contain object area, object orientation, and object appearance. These learning mechanisms could be divided into supervised learning or unsupervised learning mechanisms.

1. *supervised learning:* a sequence of labeled objects is *used as input during the training process.* The goal of the is to learn to produce the correct decision for a given new input. There are different supervised learning methods such as neural networks [91], adaptive boosting [92], decision trees [93], and support vector machines [94].
2. *unsupervised learning*: a sequence of unlabeled objects is used in training. The goal here is to build representations of the input object regions that can be used for

decision-making, predicting future candidate object. In a sense, unsupervised learning can be thought of as finding patterns in the input data and beyond what would be considered pure un-structured noise. Three examples of unsupervised learning are data clustering, data dimensionality reduction, and deep-learning.

3.4.1 Review of literature on robust tracking using deep-learning

Deep-learning have recently attracted considerable attention in the field of machine learning [95] [96], it has been successfully applied to many computer vision applications including visual object tracking [97]. Deep-learning aims to replace hand-crafted features with high-level and robust features learned from raw pixel values, which is also known as unsupervised feature learning. Many deep learning-based trackers were developed using different network models such as: SAE, *i.e., stacked auto-encoder*, CNN, *i.e., convolutional neural network*, RNN, *i.e., recurrent neural networks*, DRL, *i.e., deep reinforcement learning*, in addition to their different combinations.

In [98], the authors introduced a robust discriminative deep-learning based visual tracker by effectively using an image representation that was learned automatically. Through an offline training step, this visual tracker trained an SDAE, *i.e., stacked denoising auto-encoder*, to learn generic image features. The essential building block of an SDAE is a one-layer neural network. After the offline training step, the encoder part of the SDAE was used a feature extractor to train a neural network to identify a tracked object from the background. Eventually, both the feature extractor and classifier were further tuned to adapt to appearance changes of the moving object.

The authors in [99] used a CNN architecture in the design and implementation of a structured output deep-learning based visual tracker (SO-DLT). This implementation included an offline pre-train and online steps to fine-tune the CNN architecture. While the accuracy and performance of this visual tracker demonstrated much improvement over other state-of-the-art visual trackers, some failed visual tracking cases were reported. Such tracking failures were likely when 1) an initial object bounding box was not specified correctly, 2) distractors existed in the background, or 3) a tracked object was occluded.

Better mathematical models have been examined in [100], to eliminate or reduce the previously mentioned failed tracking cases, and to increase tracking accuracy. The authors described a robust tracker based on a CNN, trained in a multi-domain learning framework, known as a multi-domain network (MDNet). This network consists of shared layers and branches of domain-specific layers. To obtain a representation of a generic object, this tracker pre-trained the MDNet using a broad set of video sequences with known tracking ground-truths. The entire network was pre-trained offline, and the fully connected layers including a single domain-specific layer were fine-tuned online.

Most CNN-based trackers, handle visual tracking as a classification problem. But these trackers are sensitive to distractors because their CNN models mainly focus on inter-class classification. To cope with this problem, the work in [101], used self-structure information of an object to distinguish it from possible distractors. Specifically, SANet uses a recurrent neural network (RNN) to model object structure and incorporate it into the CNN to improve its robustness in the presence of distractors.

To overcome sudden changes in the appearance of objects, in [102] the authors developed a robust visual tracker based on a template selection strategy constructed by deep reinforcement learning methods to update the object's appearance model online. This tracking algorithm used this strategy to choose the best template for object tracking in a given frame. This template selection strategy was self-learned using a simple gradient method applied to many training videos that were randomly generated from a visual tracking benchmark dataset. Although this tracking algorithm effectively decided on the best object template to be used, its accuracy still needed improvement.

3.5 Chapter summary

In this chapter, three approaches to robust visual tracking in challenging conditions and unexpected events were introduced: a multi-scale approach using wavelet domain, a fusion approach, and machine learning approach. These challenging conditions and unexpected events could include 1) the presence of background motion and shadows; 2) the presence of objects with different sizes and contrast levels; 3) the presence of partial object camouflage; 4) sudden changes in scene illumination; 5) a low signal-to-noise (SNR) ratio; and 6) real-time processing requirements. A review of the literature on robust visual tracking methods using a multi-scale approach, in addition to a brief discussion of fusion, were presented.

Chapter 4

Robust Tracking of Multiple Objects in Video by Adaptive Fusion of Subband Particle Filters

4.1 Introduction

Significant progress has been made on visual tracking in the last few decades. However, the ability to track objects accurately in video sequences characterized by challenging conditions and unexpected events remains an important research problem. To address such difficulties, we developed a robust multi-scale visual tracker that represents a captured video frame as different subbands in the wavelet domain. This multi-scale tracker then applies N independent particle filters to a small subset of these subbands, with the choice of these wavelet subband subsets changing with each captured frame. Finally, the tracker fuses the outputs of these N independent particle filters to obtain the final position tracks of multiple moving objects in the video sequence. To demonstrate the robustness of our multi-scale visual tracker, we applied it to four example videos that exhibited different combinations of background motion, changes in illumination, varied object size, object shadow, and partial object camouflage. When compared to a standard full-resolution particle filter-based tracker, and a single wavelet subband, $(LL)_2$, based tracker, the results obtained from our multi-scale tracker demonstrate significantly more accurate tracking performance, as well as a reduction in average frame processing times.

This chapter is organized as follows: Section 4.2 describes the development of our robust multi-scale visual tracker. Section 4.3 provides a performance evaluation of our multi-scale tracker. Finally, Section 4.4 presents a chapter summary.

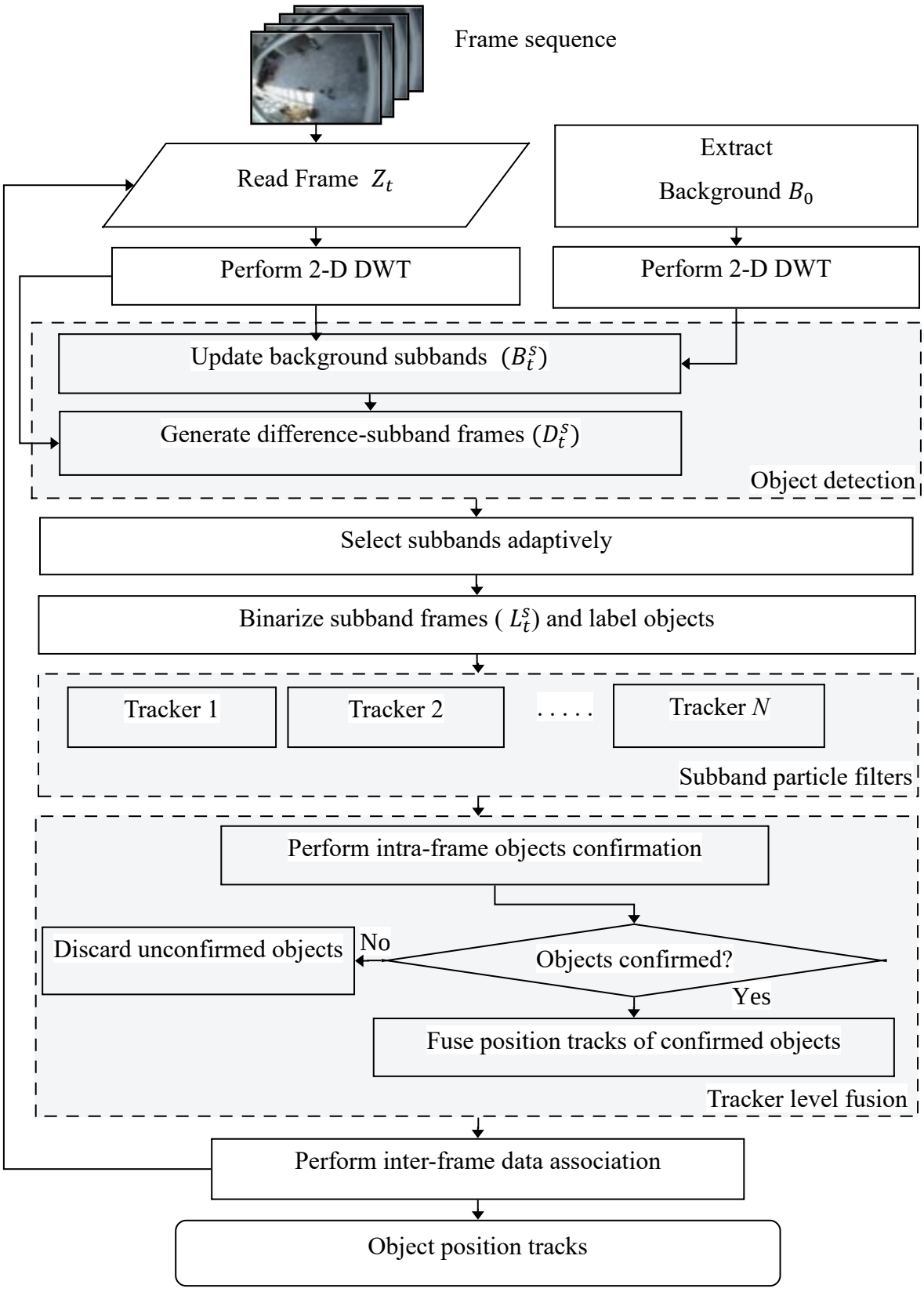


Figure 4.1. Implementation of our multi-scale visual tracker

4.2 Implementation of the Adaptive Fusion of Subband PFs multi-scale tracker

To start visual tracking, we built a *background frame* from the full-resolution video sequence. We then generated the subband frames by applying the *discrete wavelet transform* to both the background frames and the current frames. Next, we obtained *difference frames* by subtracting the subbands of the background frame from the corresponding subbands in the current frame. Once the *difference frames* had been obtained, we applied a sequential particle filter to process the three chosen subbands, which changed with each captured frame. Eventually, we fused the position tracks that were produced from the sequential processing of the three subbands to obtain our final position tracks. A flow chart demonstrating the implementation of our proposed robust visual tracker is represented in Figure 4.1.

4.2.1 Initial background extraction and update

Our first task was to detect the presence of moving objects by extracting a reference frame representing the background. To do so, we chose the *Long-Term Average Background Modeling* (LTABM) background extraction method [103], which is a fast technique that is suitable for real-time application. We obtained an initial background frame, B_0, by calculating the average of the first T video frames.

$$B_0 = \frac{1}{T}\sum_{t=0}^{T} Z_t \tag{4.1}$$

Before being used, B_0 was transformed to the wavelet domain to obtain B_0^S. The initial background subband frames each had different s scales, and at every time instant, $t > T$,

these background subband frames were updated to obtain the current background subband frames, B_t^s. We modeled each pixel (i, j) of each subband frame having scale *s* comprising, B_t^s , as an independent Gaussian probability density whose mean was updated in time as

$$B_t^s(i,j) = \alpha\, B_{t-1}^s(i,j) + (1 - \alpha)\, Z_t^s(i,j) \tag{4.2}$$

where α is an empirical weight that controls the background update rate.

4.2.2 Generation of subband frames using a multi-scale median transformation

Although many wavelet basis functions could be used to perform a wavelet transform, the selection of the proper wavelet basis is necessary. One must take into account wavelet properties such as symmetry, orthogonality, and the number of vanishing moments to enhance the detection of objects' edges at different orientations and scales. Symmetric wavelets are desirable for edge detection, as edges are obtained by differentiating a smoothed video frame. As there is also a trade-off between orthogonality and a wavelet's symmetry properties, we chose for our implementation the zbo6.6 wavelet [104], which is a symmetric biorthogonal wavelet. The zbo6.6 wavelet's decomposition filters are shown in Eq. (4.3).

$$\tilde{h} = \{0.0044, 0.223, 0.441, 0.441, 0.223, 0.0044\}$$

$$\tilde{g} = \left\{\frac{3\sqrt{2}}{16}, \frac{15\sqrt{2}}{16}, \frac{5\sqrt{2}}{4}, \frac{-5\sqrt{2}}{4}, \frac{-15\sqrt{2}}{16}, \frac{-3\sqrt{2}}{16}\right\}$$

$$h = \left\{\frac{3\sqrt{2}}{16}, \frac{-15\sqrt{2}}{16}, \frac{5\sqrt{2}}{4}, \frac{5\sqrt{2}}{4}, \frac{-15\sqrt{2}}{16}, \frac{3\sqrt{2}}{16}\right\} \tag{4.3}$$

$$g = \{0.004, -0.223, 0.441, -0.441, 0.223, -0.004\}$$

where $\tilde{h}$ and $\tilde{g}$ are the low-pass decomposition coefficients and high-pass decomposition coefficients, respectively. Because of its symmetry, obtaining a wavelet transform using the zbo6.6 wavelet could be more computationally efficient compared to other possibly asymmetric wavelets with the same support.

In visual tracking, one aims to preserve video frame edges and to reduce the effects of noise and outlier pixel values. However, there is a conflict between noise reduction and edge preservation [105]. Therefore, it has become common to use wavelet domain thresholding methods to reduce Gaussian white noise while not significantly reducing edge sharpness [106]. Instead of generating our subband frames using only a zbo6.6-based DWT, we generated them using a combination of a zbo6.6-based DWT and a multi-scale median filter [72]. This approach is advantageous because thresholding the coefficients of this *multi-scale median transformation* will still reduce white Gaussian noise, and its inherent multi-scale median filter will also reduce both speckle noise and salt and pepper noise.

4.2.3 Generation of subband difference frames, adaptive subband frame selection

At every time, $t > T$, we generated subband *difference frames*, $D_t^s(i,j) = Z_t^s(i,j) - B_t^s(i,j)$, where s belongs to the set of available subband frames in levels one and two of the wavelet tree as shown in Figure 4.2. These subband *difference frames* were generated by subtracting the current background B_t^s from the current frame Z_t^s. After calculating the L_1 norm for each subband *difference frame*, we retained the three subband *difference*

frames with the highest L_1 norm values and discarded the rest. We did not use the L_2 norm, which is the exact signal energy; instead, we used the L_1 norm, or the approximation of the signal energy, in order to avoid the additional computational cost as shown in Eq. (4.4).

$$E(D_t^s) = |D_t^s|_1 = \sum_{i}^{I} \sum_{j}^{J} |D_t^s(i,j)| \tag{4.4}$$

Since features of a moving object such as size, shape, and orientation affect the energy distribution of the subband frames, this adaptive selection of the *difference frames* would be matched to the features of the moving object at each time instance $t > T$. For example, if the orientation of a moving object is horizontal, then the energy of the (LH) subband *difference frame* will be higher than all others. In contrast, if the orientation of a moving object is vertical, then the energy of the (HL) subband *difference frame* will be the higher than all others. This was observable when the chosen subbands were (HL), and (LH).

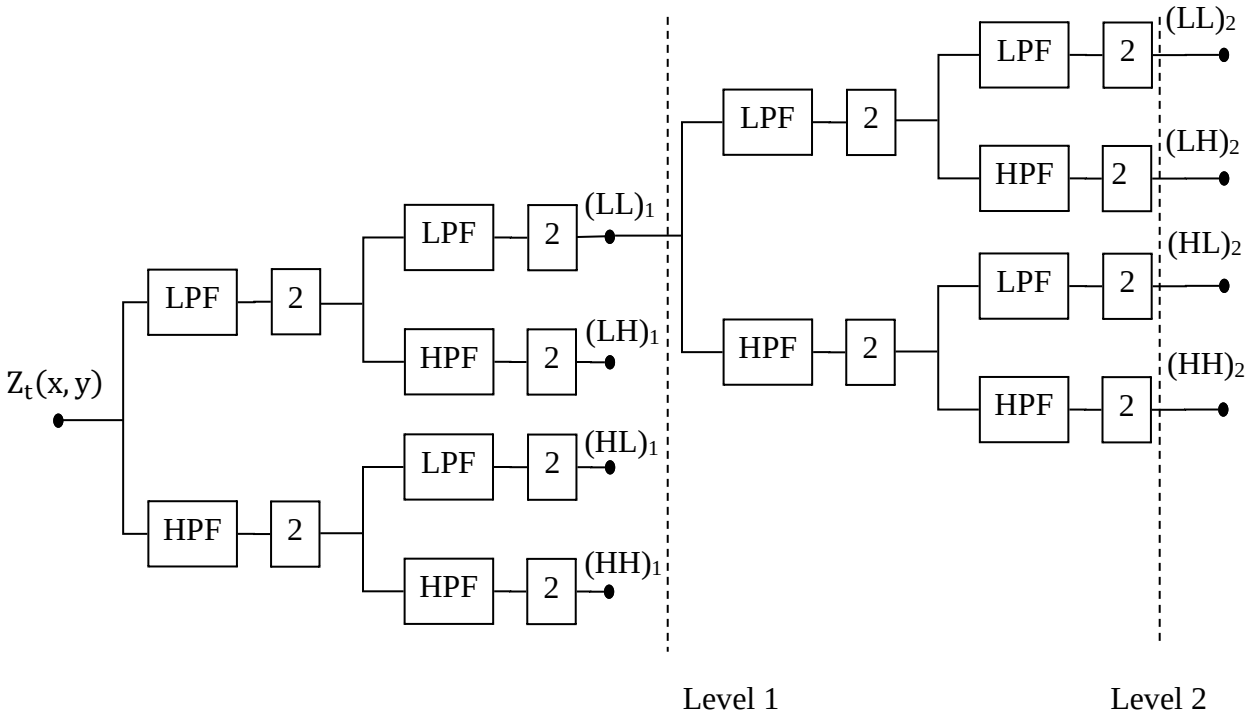


Figure 4.2. Subband frames from level 1 and 2 used in our tracker

4.2.4 Frame denoising

A robust tracker should have the ability to cope with video sequences that have a low signal-to-noise ratio. The regular median filter is better than linear filters for removing noise while preserving the presence of edges [107]. However, the window size of the regular median filter affects its ability to preserve edges [108]. Fortunately, a multiscale median filter could better preserve image details. Therefore, we selected a multi-scale median transform for denoising the video frames. We combined wavelet transformation and multi-scale median filtering, will inherently reduce both speckle noise and salt and pepper noise. By only keeping the three subband *difference frames* with the highest energies, we reduced the white Gaussian noise originally present in the full-resolution

frame via hard thresholding.

4.2.5 Frame binarization and object labeling

Both frame binarization and object labeling step aim to create labeled groups of pixel regions that contain candidate moving objects in a subband frame. Labeling is especially necessary when the scene contains more than one moving object [109]. We generated *binary frames* from the three selected *difference frames* via thresholding, where pixels with values above a positive threshold, *k*, were categorized as foreground. This is illustrated in the below inequality,

$$|D_t^s|_{threshold} = |Z_t^s - B_t^s| > k \tag{4.5}$$

where $|D_t^s|_{threshold}$ are the binary frames and d refers to the threshold level. The resulting white pixels in the binary frame would then refer to candidate moving objects. After generating these binary frames, morphological operations that include dilation and fill operations were implemented to enhance the shapes of present objects. A *labeled frame* was then obtained by scanning the *binary frame*, pixel by pixel from left to right and top to bottom, to identify the present connected pixel regions.

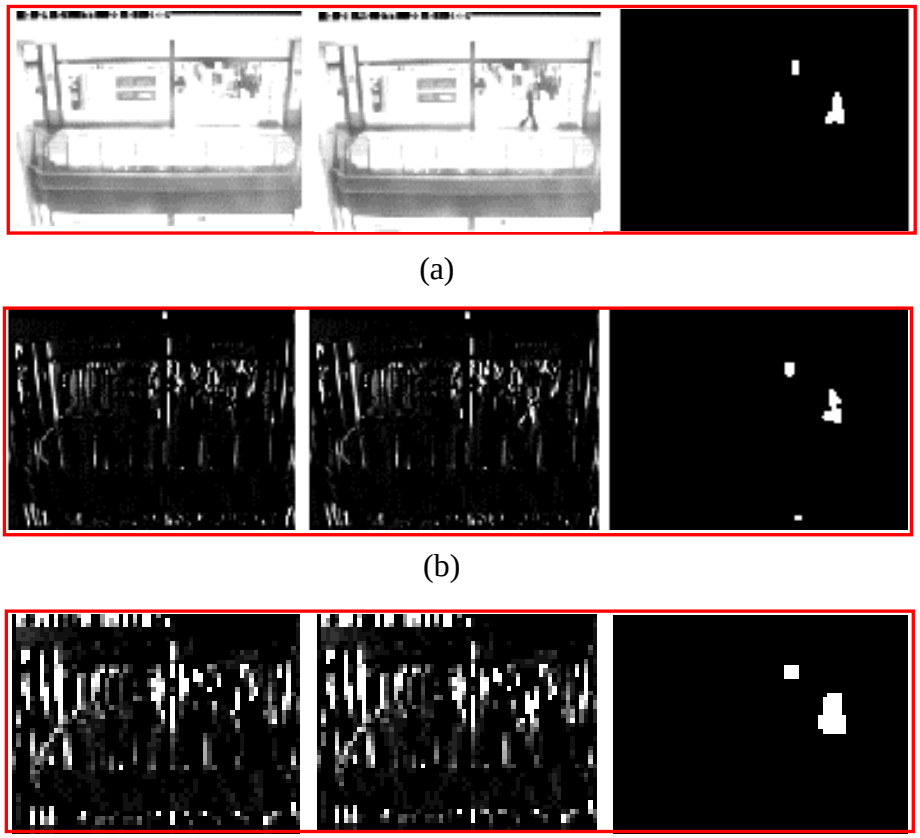


Figure 4.3. Result of subband generation and frame binarization for the adaptively chosen subbands $(LL)_1$, $(HL)_1$, and $(HL)_2$ for the 4th frame in *''OneLeaveShopReenter2front''* video sequence. (a) Background, current, and binary frames, respectively, for the $(LL)_1$ subband (upper row); (b) Background, current, and binary frames, respectively, for the $(HL)_1$ subband (middle row); and (c) Background, current, and binary frames, respectively, for $(HL)_2$ subband (bottom row)

Figure 4.3 shows the result of frame binarization of the 4th frame in *''OneLeaveShopReenter2front''* video sequence. For this particular frame, the chosen subbands were $(LL)_1$, $(HL)_1$, and $(HL)_2$, and the orientation of the moving objects was vertical. Therefore, the energy of the (HL) subband *difference frame* was the higher than the others as is demonstrated in Section 5.3.

4.2.6 Implementation of our subband particle filters

We implemented three independent particle filters, where each filter processed one of the three labeled binary frames obtained in Section 4.2.5. These subband particle filters continuously updated the kinematic states of the objects present in these labeled binary frames. We note that we used a linear motion model similar to the one described in [110], and a measurement model based on motion cues similar to the one described in [111].

4.2.6.1 Likelihood model

Typically, a likelihood model uses a visual feature, e.g., color, shape, texture, edge, or motion. In the case of a static camera, it is common to use motion cues [112]. In our visual tracker, the likelihood model used a motion cue. Hence, instead of using the pixels of the captured frame to generate the likelihood function, we instead used the binary *labeled frame*, L_t^s. As with the work in [111], we defined our likelihood model by evaluating white pixels associated with each object n and belonging to a boundary box R located around spatial position u of state the vector X_t. Thus, the higher the number of labeled pixels contained in the boundary box, the higher likelihood score. This relationship is shown in Eq. (4.6).

$$p(L_t^s|X_t) \propto \sum_{n=1}^{N} w_n$$

$$w_n = \sum_{i,j \in R} L_t^s(i,j) \tag{4.6}$$

where i and j are the spatial coordinates of a white pixel inside the boundary box R associated with object n where $n \in \{1, .., N\}$ and N is the number of objects.

4.2.6.2 Prior motion model

The motion model describes the temporal evolution of the system state statistics $X_t = (x_t^1, x_t^2, \dots, x_t^N)^T$. The vector x_t^n can be expressed as $x_t^n = [u_t^n, v_t^n]^T$ where u_t^n and v_t^n are the position and the velocity of the object n. A linear motion model describes the temporal evolution as shown in Eq.(4.7).

$$u_{t+\delta t}^l = u^l{}_t + v_t^l \, \delta t + f$$

$$v_{t+\delta t}^l = v_t^l \; + w \tag{4.7}$$

where δt is the time step between successive frames, and f and w are the excitation forces modeled by a uniform variable in a certain range, which relates to expected change in the object's position and velocity.

4.2.7 Fusion of position tracks from our subband particle filters

Our subband particle filters produced three sets of position tracks corresponding to multiple moving objects. To obtain the final set of position tracks, we performed an object confirmation step (explained in Section 5.7.1), which was followed by an averaging of the confirmed objects' position tracks.

4.2.7.1 Intra-frame Object confirmation

Object confirmation is an intra-frame data association step wherein the presence of an object in a predefined region is asserted by majority voting. If the majority of our subband particle filters agreed that there was an object in a predefined region, then this

object would be confirmed. As a result, phantom objects that could be falsely detected and tracked by a minority of subband particle filters would be discarded.

4.2.8 Inter-frame Data association

To preserve consistent identities of tracked objects over time, we performed one, or possibly two, inter-frame data association steps. The first step, *position gating*, imposes a constraint that an object i at time t can be associated with an object j at time $t-1$ if the distance between them is less than a defined gate size.

4.2.8.1 Inter-frame Data association

If the *position gating* step failed to associate an object, i, at time t with an object, j, at time $t-1$, we resorted to a *gray-scale histogram comparison* step. We started by matching the area, A_i^0, of the unidentified object, i, in the full-resolution frame to its area, A^s, in our chosen subband frames. We then evaluated a normalized histogram of the gray levels, $\widetilde{q_i}$, in the area, A_i^0, in addition to a normalized histogram of the gray levels, $\widetilde{q_j}$, of each object j in the previous frame. Then, a similarity measure, d, based on the Bhattacharyya distance, p, between the two normalized histograms distributions was computed as,

$$p = \sum_{b=1}^{B} \sqrt{\widetilde{q_i}(b)\widetilde{q_j}(b)}$$

$$d = \sqrt{1-p} \tag{4.8}$$

where b is an index representing the B histogram bins used. Based on the value of the distance d between an object, i, at time t and an object, j, at time $t-1$, the current object was assigned the identity of one of the objects present at $t-1$, or it was assigned a new object identity.

4.3 Performance evaluation of our robust multi-scale visual tracker

In the following examples, we show that our visual tracker overcame the presence of challenging conditions in four video sequences. Moreover, we show that our tracker demonstrated better tracking performance compared to a typical visual tracker using a standard full-resolution particle filter-based and single wavelet subband $(LL)_2$ based tracker.

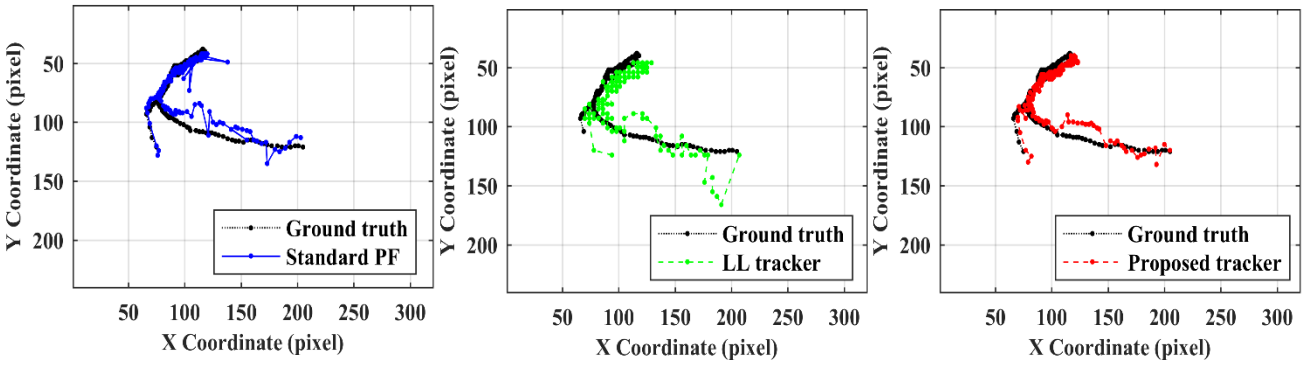
4.3.1 Example demonstrating partial object camouflage and object shadow

To demonstrate the improved performance of our multi-scale subband particle filters tracker, we applied it to an *"Intelligentroom_raw"* video sequence that included the presence of object shadow and partial object camouflage. This video sequence depicted a man walking around a conference room. The true position track of the object, i.e., our ground truth, was also available via the VISOR database.

4.3.1.1 Comparison of resulting position tracks

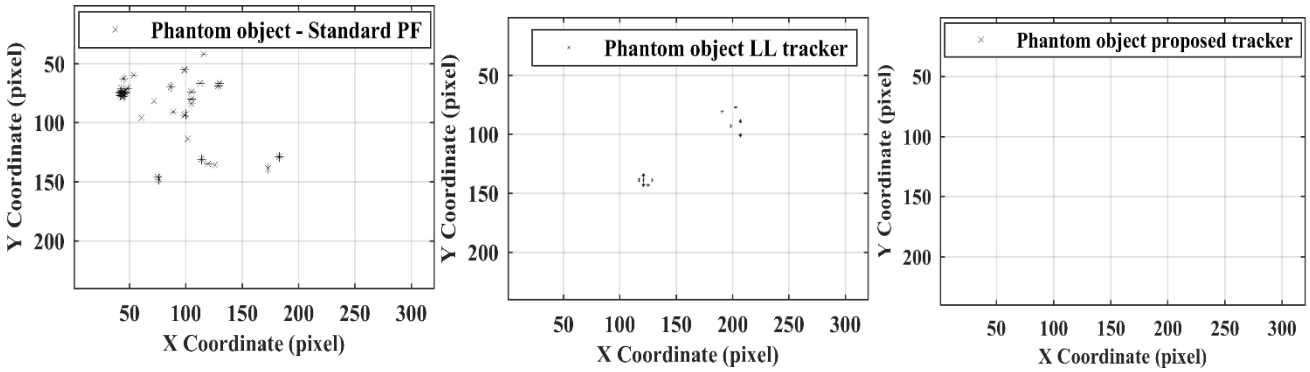
Figure 4.4 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ tracker, and our multi-scale subband particle filter tracker. Figure 4.4 (a), Figure 4.4 (b), and Figure 4.4 (c) show the true position tracks of the object, as well as those generated by the standard full-resolution

particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are significantly smaller than when a standard full resolution particle filter-based tracker was used.



(a) (b) (c)

Figure 4.4. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) a single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker



(a) (b) (c)

Figure 4.5. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

Figure 4.5 shows the position tracks of phantom objects generated by the standard full-resolution particle filter-based tracker and the single wavelet subband $(LL)_2$ based tracker. These phantom objects may have appeared due to the presence of the object's shadow or partial object camouflage. We note that our multi-scale tracker generated no phantom objects, while the standard full-resolution particle filter-based tracker and the single wavelet subband $(LL)_2$ based tracker generated many.

To quantitatively compare the performance of these three visual trackers, we will define a *detection frame* of a specific object as a frame where this particular object was correctly detected by these three trackers. As shown in Table 4.1, the object in this video appeared in 214 *detection frames*, with cumulative track errors of 982 pixels, 2024 pixels, 1025 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively.

We note that due to the presence of challenging conditions and unexpected events in this video sequences, .e.g., object shadow; partial object camouflage; and low signal-to-noise ratio, standard full resolution particle filter-based tracker generated 90 phantom objects, also the single wavelet subband $(LL)_2$ based tracker generated 11 phantom objects, while our multi-scale tracker overcame the presence of these challenging conditions and unexpected events and generated no phantom objects. These values are a solid demonstration of the superior performance and robustness of our multi-scale tracker compared to the other two trackers.

Table 4.1. Number of missed object events, average position track errors, and number of phantom object events

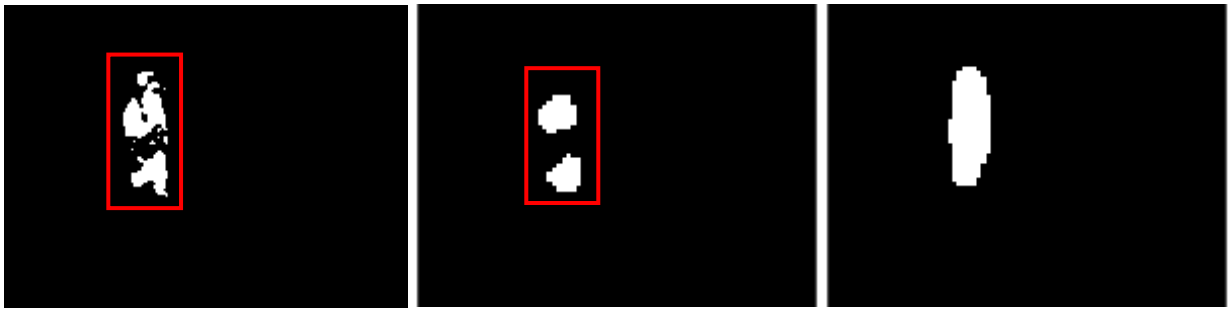
Visual tracker type	Missed object (event/300 frames)	Average position track error (pixel/*detection frame*)	Standard deviation of tracking errors	Phantom object (event/300 frames)
Full resolution particle filter tracker	0	4.635	5.5	90
$(LL)_2$ subband tracker	2	9.547	5.8	11
Our multi-scale tracker	0	4.8	2.86	0

4.3.1.2 Demonstrating challenging video conditions

Partial object camouflage: Figure 4.6 (a), Figure 4.6 (b), and Figure 4.6 (c) show the binary frames generated from the 265^{th} video frame using the full-resolution frame, subband $(LL)_2$, and subband $(HL)_2$ — that is, one of the three chosen in our multi-scale tracker— respectively. We note that the red boxes in Figure 4.6 (a), and Figure 4.6 (b) highlight the division of an object into two due to partial object camouflage.

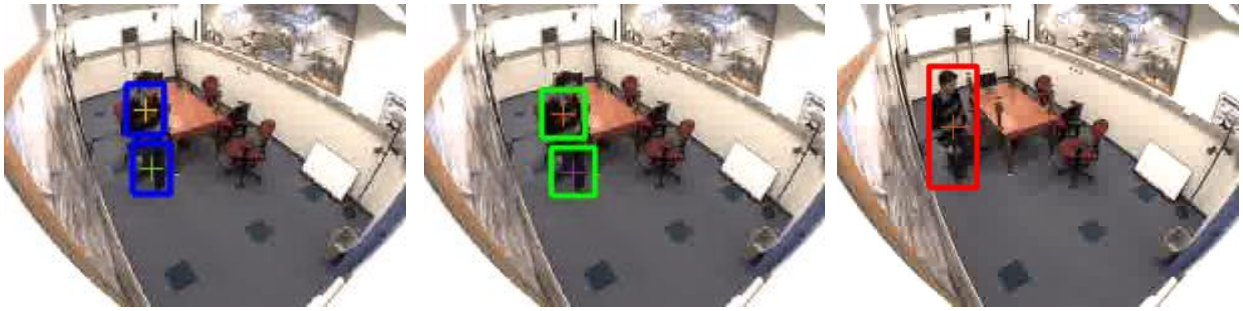
Figure 4.7 (a), Figure 4.7 (b), and Figure 4.7 (c) show visual tracking results, superposed onto the 265^{th} video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker and our multi-scale tracker, respectively. We note that the standard full-resolution particle filter-based tracker

and single wavelet subband $(LL)_2$ based tracker generated two phantom objects due to the object division in Figure 4.6 (a), and Figure 4.6 (c), while our multi-scale tracker overcame the presence of partial object camouflage in the 265th video frame.



(a) (b) (c)

Figure 4.6. Binary frames generated from the 265th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$



(a) (b) (c)

Figure 4.7. Visual tracking results for 265th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

Object shadow: Figure 4.8 (a), Figure 4.8 (b), and Figure 4.8 (c) show the binary frames generated from the 100th video frame using the full-resolution frame, subband $(LL)_2$, and subband $(HL)_1$ — that is, one of the three chosen subbands for this frame in our proposed multi-scale tracker—, respectively.

We note that the red box in Figure 4.8 (a) highlights an artifact due to the presence of object shadow. Figure 4.9 (a), Figure 4.9 (b), and Figure 4.9 (c) show visual tracking results, superposed onto the 100^{th} video frame, generated by a standard full-resolution particle filter-based tracker, a single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 4.8 (a), while our multi-scale tracker and the single wavelet subband $(LL)_2$ based tracker overcame the presence of object shadow in the 100^{th} video frame.

(a) (b) (c)

Figure 4.8. Binary frames generated from the 100th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$

(a) (b) (c)

Figure 4.9. Visual tracking results for 100th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

4.3.2 Example demonstrating background motion, object shadow, and partial object camouflage

To demonstrate the improved performance of our multi-scale subband particle filter tracker, we applied it to the *"OneLeaveShopReenter2front"* video sequence that included background motion, object shadow, and partial object camouflage. In this video sequence example, two people walk past the front of a store, while another person exits the store and then re-enters. The true position tracks of these objects, i.e., our ground truth, were also available via the CAVIAR database.

4.3.2.1 Comparison of resulting position tracks

Figure 4.10 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker. Figure 4.10 (a)-(c), Figure 4.10 (d)-(f), and Figure 4.10 (g)-(j) show the true position tracks of the three objects, as well as those generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively.

We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are significantly smaller than the differences generated by the standard full-resolution particle filter-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 55, 80, and 23 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively.

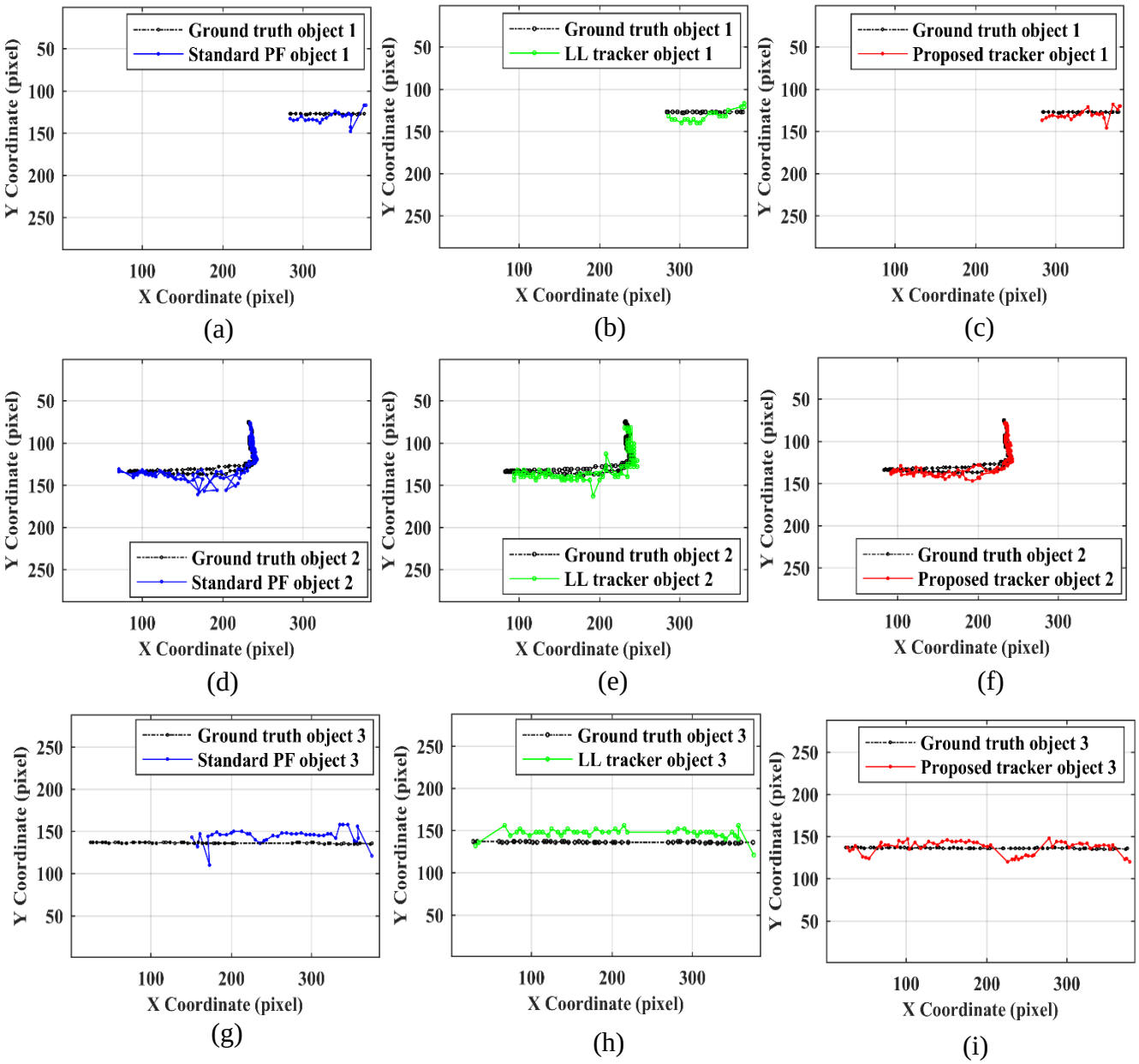


Figure 4.10. Position tracks of true objects (a) - (i) in the *''OneLeaveShopReenter2front''* video using a standard full-resolution particle filter-based tracker (right column), an LL-based tracker (middle column), and our multiscale tracker (left column)

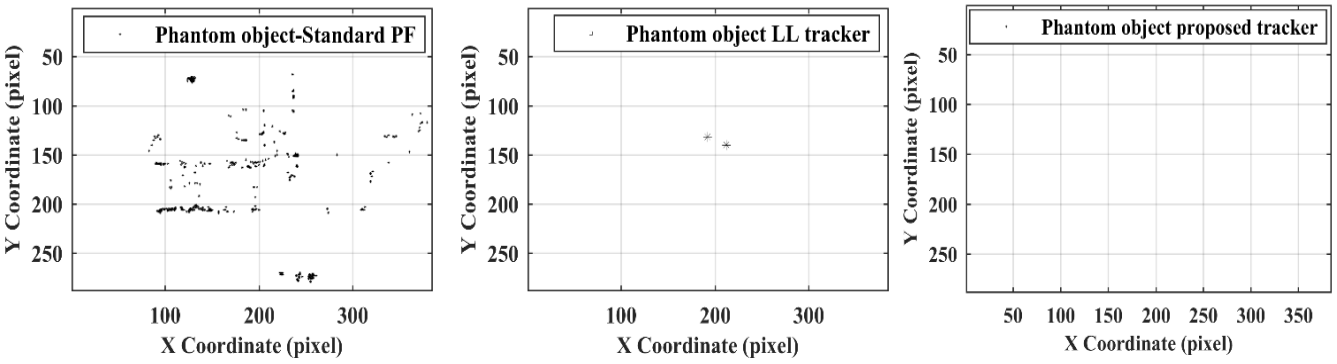


Figure 4.11. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our proposed multiscale

Figure 4.11 shows the position tracks of phantom objects generated by the standard full-resolution particle filter-based tracker. These phantom objects may have appeared due to the presence of background motion, object shadows, or partial object camouflage. We note that our multi-scale tracker generated no phantom objects, while the standard full-resolution particle filter-based tracker generated many and the single wavelet subband $(LL)_2$ based tracker generated two. This disparity further demonstrates the robustness of our multi-scale tracker.

As shown in Table 4.2, object 1 in this video appeared in 58 *detection frames*, with cumulative track errors of 381 pixels, 474 pixels, 309 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively. Object 2 in this video appeared in 470 *detection frames*, with cumulative track errors of 1876 pixels, 3407 pixels, 2149 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively. Object

3 in this video appeared in 121 *detection frames*, with cumulative track errors of 1595 pixels, 1510 pixels, 805 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively. These values are a solid demonstration of the superior performance and robustness of our multi-scale tracker compared to the other two trackers.

Table 4.2. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker	Missed object (event/558 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/558 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	55	6.56	3.99	13.18	3.6	3.12	5.58	469
$(LL)_2$ subband tracker	80	8.17	7.24	12.47	3.84	2.97	3.41	2
Our multi-scale tracker	23	5.32	4.57	6.65	2.68	2.37	2.82	0

4.3.2.2 Demonstrating challenging video conditions

Background motion: Figure 4.12 (a), Figure 4.12 (b), and Figure 4.12 (c) show the binary frames generated from the 6th video frame using the full-resolution frame, subband $(LL)_2$, and subband $(HL)_2$ — that is, one of the three chosen subbands for this frame in our multi-scale tracker— respectively.

(a) (b) (c)

Figure 4.12. Binary frames generated from the 6th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_2$

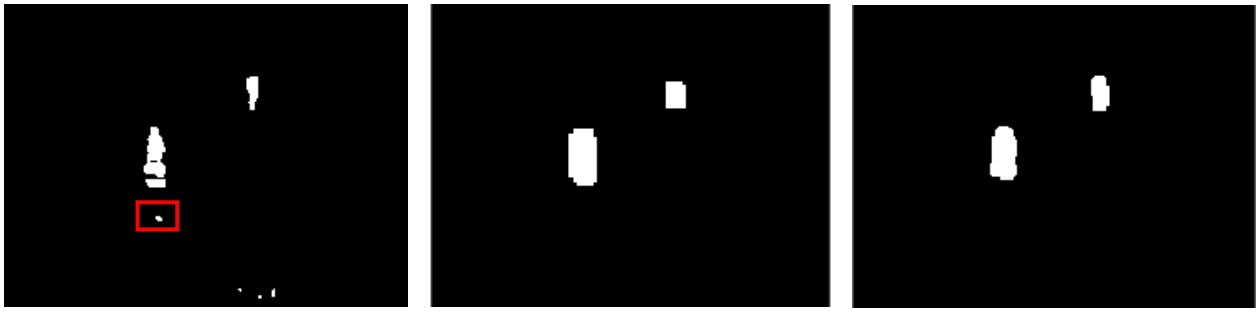
(a) (b) (c)

Figure 4.13. Visual tracking results for the 6th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

We note that the red box in Figure 4.12 (a) highlights an artifact due to the presence of background motion. Figure 4.13 (a), Figure 4.13 (b), and Figure 4.13 (c) show visual tracking results, superposed onto the 6th video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 4.13 (a) and the single wavelet subband $(LL)_2$ based tracker lost one object as it used only

one subband at fixed scale (LL_2); conversely, our multi-scale tracker succeeded in tracking both objects and overcame the presence of background motion in the 6th video frame.

Object shadow: Figure 4.14 (a), Figure 4.14 (b), and Figure 4.14 (c) show the binary frames generated from the 116th video frame using the full-resolution frame, subband $(LL)_2$, and subband $(HL)_1$ — that is, one of the three chosen subbands for this frame in our multi-scale tracker— respectively. We note that the red box in Figure 4.14 (a) highlights an artifact due to the presence of object shadow.



(a) (b) (c)

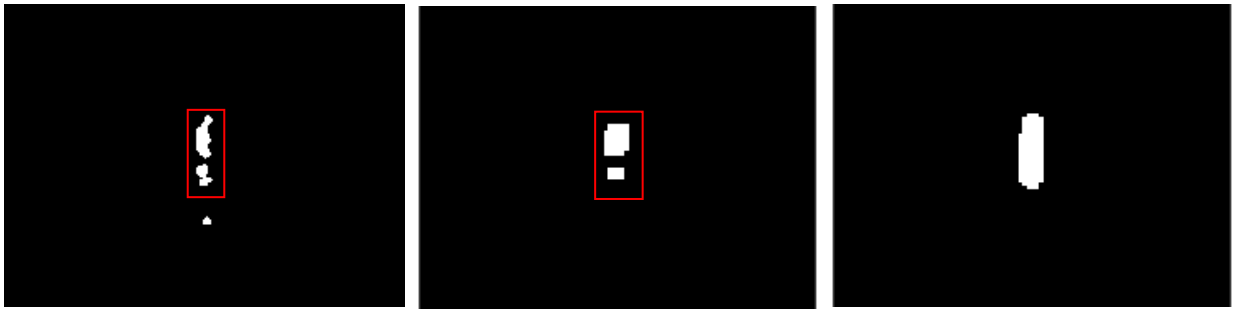
Figure 4.14. Binary frames generated from the 116th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$

(a) (b) (c)

Figure 4.15. Visual tracking results for the 116th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

Figure 4.15 (a), Figure 4.15 (b), and Figure 4.15 (c) show the visual tracking results, superposed on the 116th video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 4.14 (a), while our multi-scale tracker overcame the presence of object shadow in this 116th video frame.

Partial object camouflage: Figure 4.16 (a), Figure 4.16 (b), and Figure 4.16 (c) show the binary frames generated from the 427th video frame using the full-resolution frame, subband $(LL)_2$, and subband $(HL)_2$ — that is, one of the three chosen subbands for this frame in our multi-scale tracker— respectively. We note that the red box in Figure 4.16 (a), and Figure 4.16 (b) highlights the division of an object into two due to the presence of partial object camouflage.



(a) (b) (c)

Figure 4.16. Binary frames generated from the 427th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_2$

Figure 4.17 (a), Figure 4.17 (b), and Figure 4.17 (c) show the visual tracking results, superposed onto the 427th video frame, generated by the standard full-resolution particle filter-based tracker and our multi-scale tracker, respectively. We note that standard full-resolution particle filter-based tracker generated two phantom objects due to the object division in Figure 4.16 (a) and the object shadow, and the single wavelet subband $(LL)_2$ based tracker generated a phantom object due to the object division in Figure 4.16 (b). However, our multi-scale tracker was able to overcome the presence of partial object camouflage in the 427th video frame.

(a) (b) (c)

Figure 4.17. Visual tracking results for the 427th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

4.3.3 Example demonstrating a sudden change in illumination and presence of objects with different sizes

The video sequence in this example, *"Meet_WalkTogether2"*, is from the CAVIAR database. In this video sequence, two people meet and walk together. The true position tracks of these objects, i.e., our ground truth, were also available via the CAVIAR database.

4.3.3.1 Comparison of resulting position tracks

Figure 4.18 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale subband particle filters tracker. Figure 4.18 (a) - (c), Figure 4.18 (d) - (f), and Figure 4.18 (g) - (i) show the true position tracks of the three objects, as well as those generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the differences between the position paths generated by our multi-scale tracker and the true position paths

are significantly smaller than the differences generated by the standard full-resolution particle filter-based tracker and single wavelet subband $(LL)_2$ tracker.

Figure 4.19 shows the position tracks of phantom objects generated by the standard full-resolution particle filter-based tracker. These phantom objects may have appeared due to the presence of background motion, object shadows, or partial object camouflage. We note that our multi-scale tracker generated no phantom objects, while the standard full-resolution particle filter-based tracker generated many. This is a further demonstration of our multi-scale tracker's robustness.

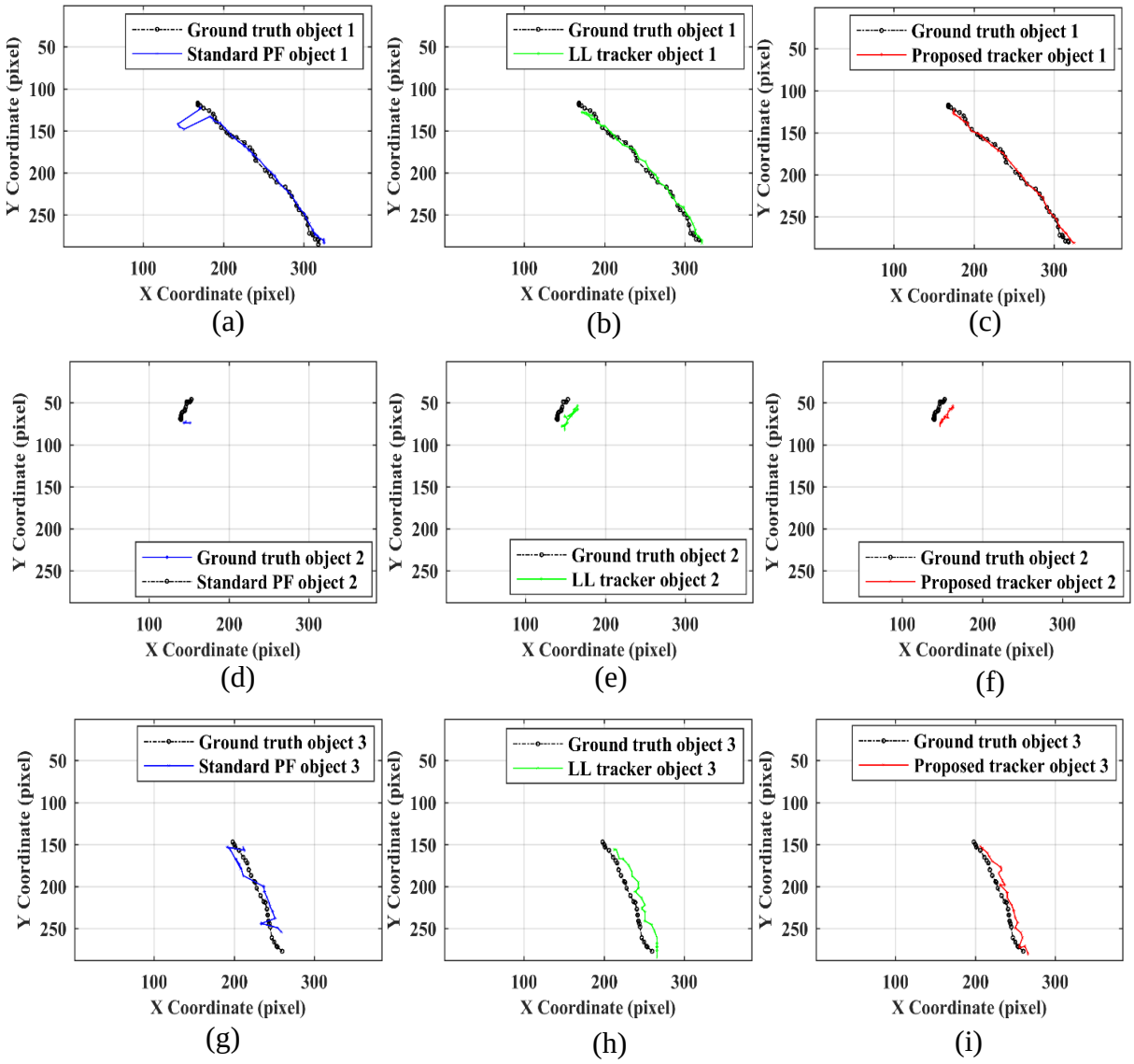
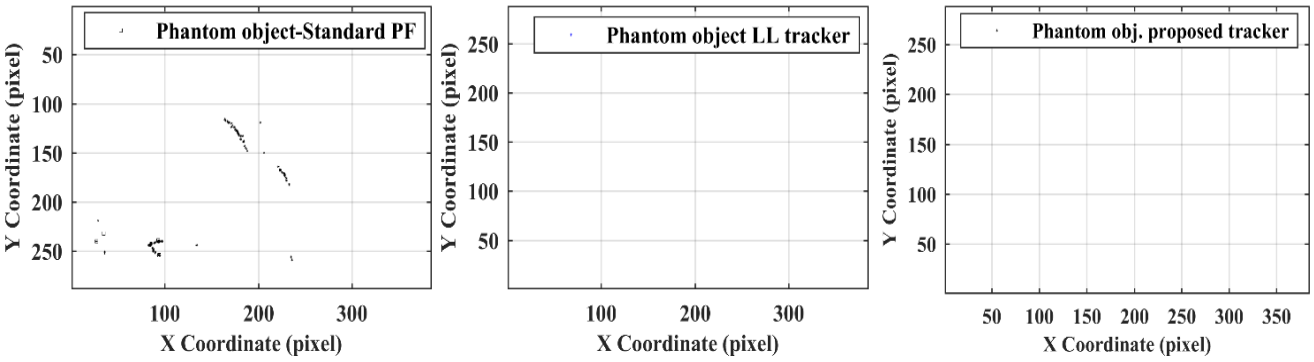


Figure 4.18. Position tracks of true objects (a) - (i) in the *"Meet_WalkTogether2"* video using a standard full resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multiscale tracker (left column)



(a) (b) (c)

Figure 4.19. Position tracks of phantom objects generated by: (a) the standard full resolution particle filter-based tracker, (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

Table 4.3 shows that object 1 in this video appeared in 109 *detection frames*, with cumulative track errors of 721 pixels, 738 pixels, 547 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively. Object 2 in this video appeared in 8 *detection frames*, with cumulative track errors of 72 pixels, 101 pixels, 80 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively. Object 3 in this video appeared in 60 *detection frames*, with cumulative track errors of 718 pixels, 1023 pixels, 653 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively.

These values are a solid demonstration of the superior performance and robustness of our multi-scale tracker compared to the other two trackers.

Table 4.3. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker	Missed object (event/827 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/827 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	81	6.6	9.04	12	7.94	3.91	3.19	122
$(LL)_2$ subband tracker	62	6.7	12.7	17.05	451	2.45	3.70	0
Our multi-scale tracker	45	5.0	10.05	10.8	3.52	0.69	3.39	0

We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than those generated by the standard full-resolution particle filter-based tracker and the LL-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 81, 62, and 45 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively.

4.3.3.2 Demonstrating challenging video conditions.

Sudden change in illumination: Figure 4.20 (a), Figure 4.20 (b), and Figure 4.20 (c) show the binary frames generated from the 56^{th} video frame using the full-resolution frame,

subband $(LL)_1$, and subband $(LH)_2$ — that is, one of the three chosen subbands for this frame in our multi-scale tracker— respectively.

(a) (b) (c)

Figure 4.20. Binary frames generated from the 56th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(LL)_1$

(a) (b) (c)

Figure 4.21. Visual tracking results for the 56th video frame using: (a) the standard full-resolution particle filter-based tracker; (b)) the single wavelet subband $(LL)_2$ based tracker; and (c) our multi-scale tracker

We note that the red box in Figure 4.20 (a) highlights an artifact due to a sudden illumination change in the video frame. Figure 4.21(a), Figure 4.21 (b), and Figure 4.21 (c) show the visual tracking results, superposed on the 56th video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$

based tracker, and our multi-scale tracker, respectively. We note that the standard full-resolution particle filter-based tracker generated a phantom object due to the artifact in Figure 4.20 (a), while our multi-scale tracker overcame the effect of the sudden change in illumination in the 56th video frame.

(a) (b) (c)

Figure 4.22. Binary frames generated from the 201st frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(LH)_2$

(a) (b) (c)

Figure 4.23. Visual tracking results for the 201st video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale tracker

Presence of objects with different sizes: Figure 4.23 (a), Figure 4.23 (b), and Figure 4.23 (c) show the binary frames generated from the 201st video frame using the full-

resolution frame, subband $(LL)_2$, and subband $(LH)_2$ — that is, one of the three chosen subbands for this frame in our multi-scale tracker— respectively. We note that the object sizes in Figure 4.23 (c) are closer to each other than the object sizes in Figure 4.23 (a).

Figure 4.23 (a), Figure 4.23 (b), and Figure 4.23 (c) show the visual tracking results, superposed onto the 201st video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that, due to the presence of a large object, the standard full-resolution particle filter-based tracker not only failed to track the smaller object, but it also exhibited a problem with partial camouflage. Also, the single wavelet subband $(LL)_2$ based tracker failed to track the small object due to using only one subband in a fixed scale: the second scale. Conversely, our multi-scale tracker was able to overcome these problems and successfully tracked both the large and small objects.

4.3.4 Example demonstrating presence of objects with different sizes and partial object camouflage

The The video sequence in this example, "ATCS" is from the Visor database (288 X 384 pixels, 30 fps, 1313 frames). This video sequence shows three moving people.

4.3.4.1 Comparison of resulting position tracks

Figure 4.25 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale subband particle filters tracker Figure 4.25 (a) - (c), Figure 4.25 (d) - (f), and Figure 4.25 (g) - (i) show the true position tracks of the three objects, as well as those generated

by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are significantly smaller than the differences generated by the standard full-resolution particle filter-based tracker and single wavelet subband $(LL)_2$ tracker.

Figure 4.24 shows the visual tracking results for a sample of four video frames using our multi-scale tracker. We note that our multi-scale tracker generated no phantom objects, while the standard full-resolution particle filter-based tracker generated many. This is a further demonstration of our multi-scale tracker's robustness.

Table 4.4 shows that object 1 video appeared in 385 *detection frames*, with cumulative track errors of 2177 pixels, 5320 pixels, 3893 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale tracker, respectively. Object 2 in this video appeared in 156 *detection frames*, with cumulative track errors of 1446 pixels, 2456 pixels, 1892 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale tracker, respectively. Object 3 in this video appeared in 289 *detection frames*, with cumulative track errors of 1621 pixels, 3730 pixels, 2463 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale tracker, respectively. These values are a solid demonstration of the superior performance and robustness of our multiscale tracker compared to the other two trackers.

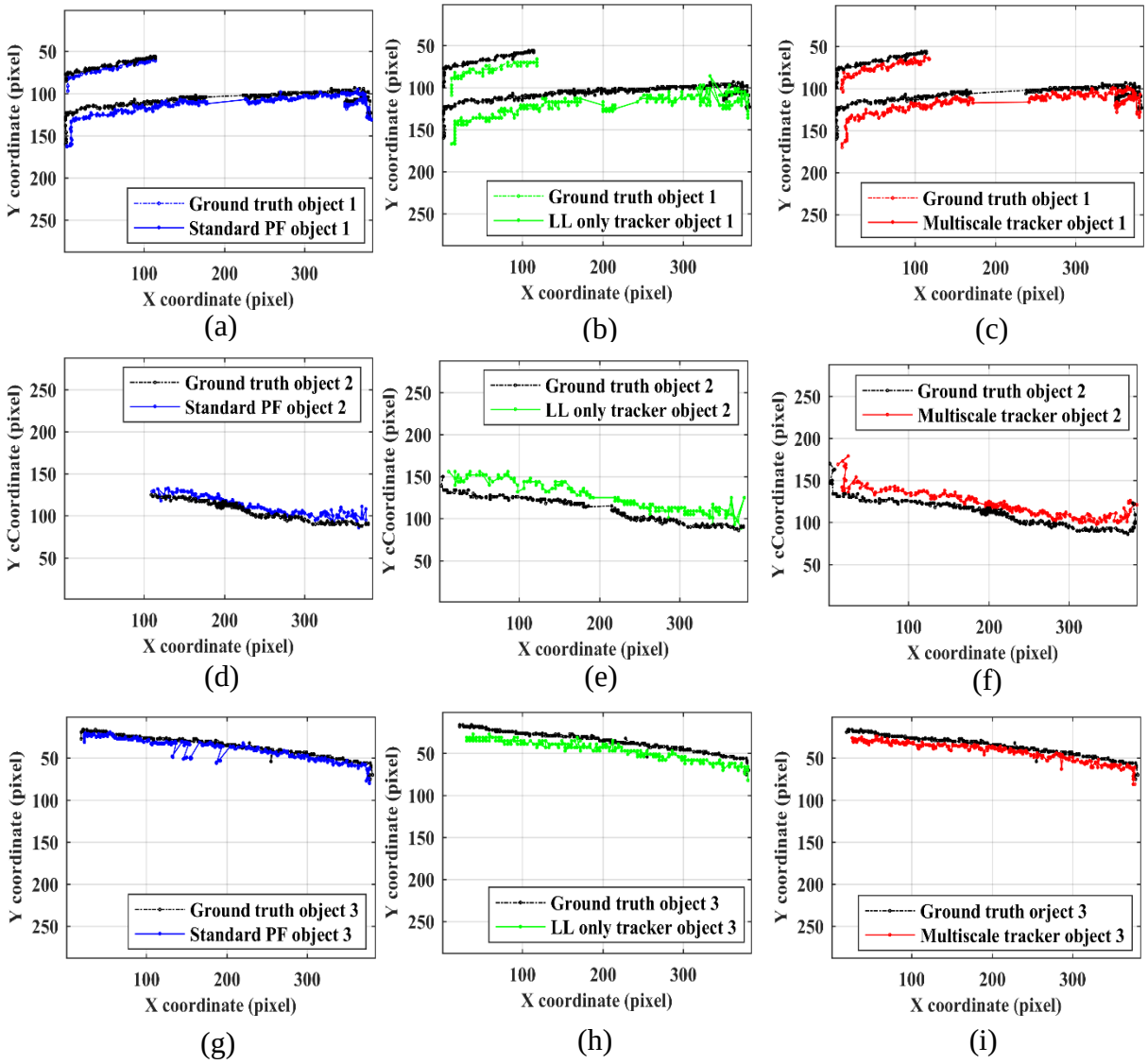


Figure 4.25. Position tracks of true objects (a) - (i) in the "*ATCS*" video using a standard full resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multiscale tracker (left column)

Figure 4.24. Visual tracking results for four video frames using our multi-scale tracker

We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than those generated by the standard full-resolution particle filter-based tracker and the LL-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 83, 34, and 15 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively.

Table 4.4. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/1313 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/1313 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	83	5.65	7.34	5.61	2.95	3.60	4.53	31
$(LL)_2$ subband tracker	34	13.81	15.76	12.90	5.03	4.59	3.90	0
Our multi-scale tracker	15	10.11	12.13	8.52	3.91	3.49	3.26	0

4.3.5 Comparison with correlation filter based visual tracker

Recently, visual trackers based on correlation filters have gained considerable attention, as these trackers achieved 1) fast real-time tracking performance, i.e., low computation cost, and 2) robust tracking performance.

However, these correlation filter based visual trackers are challenged when dealing with 1) object scale variations, i.e., change in the size of the object, and 2) partial or full object occlusion [113-115]. In comparison, where our multi-scale visual tracker demonstrated excellent performance in the presence of partial object camouflage, and object scale changes, as shown in Section 4.3.2 - example 2.

A correlation filter estimates a similarity measure between two objects by evaluating the inner product for each potential alignment using a learned object template, also known as *filter template*. Using the Fast Fourier Transform, the computation cost of correlation filter-based visual trackers could be considerably reduced.

Different modifications were applied to a correlation filter based visual tracker to address the limitation above. For example, to handle object scale variations, an adaptive multi-scale correlation filter was introduced in [116] using learned templates that were represented as an image pyramid. Furthermore, the correlation filter was combined with particle filter in the implementation of the robust tracker developed in [115]. These multi-scale and Bayesian approaches used in correlation filters based visual trackers further confirms the validity and importance of our multi-scale Bayesian based visual trackers that are described in this thesis.

4.3.6 Practical applicability and average frame processing times

In addition to demonstrating the robustness of our multi-scale tracker, we confirmed its practical applicability by examining its average frame processing times. Even though we increased the number of independent subband particle filters, the number of pixels in each subband decreased geometrically as the scale increased.

Table 4.5. Average frame computation times using single wavelet subband $(LL)_2$ based tracker, a standard full- resolution particle filter-based tracker, and our proposed tracker

Video sequences	Data-base	Standard PF tracker (sec / frame)		Single wavelet subband $(LL)_2$ based tracker (sec/ frame)		Multi-scale tracker (sec/ frame)	
		1000 Particle	3000 Particle	1000 Particle	3000 Particle	1000 Particle	3000 Particle
Intelligentroom_raw	Visor	0.0377	0.0438	0.0179	0.0193	0.0370	0.0379
OneLeaveShopReenter 2front	Caviar	0.0506	0.0598	0.0205	0.0342	0.0505	0.0582
Meet_WalkTogether2	Caviar	0.0523	0.067	0.020	0.0283	0.0509	0.0556
Atcs	Visor	0.0512	0.0552	0.0259	0.0287	0.0495	0.0549

We obtained our results using Matlab R2016 running on a 2.6 GHz Intel R Core™ i7 with 16 GB RAM. In Table 4.5, we compare the average frame processing times for a standard tracker that utilizes a full-resolution frame, a single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker.

We note that the average frame processing times for our multi-scale tracker were always comparable or better than those of the standard full-resolution particle filter-based tracker. Because of the independence of our subband particle filters, they could also be

implemented on a parallel computer to further reduce computational times, and could, therefore, be comparable to the single wavelet subband $(LL)_2$ based tracker that uses only one subband. Further, Table 4.6 shows the computational complexities of the different steps used in our multi-scale particle filter visual tracker using *N* subbands. We note that the overall computational complexity of our multi-scale *N* subband particle filter based tracker is comparable to the computational complexity of a full resolution particle based visual tracker.

Table 4.6. Computational complexity of our multi-scale *N* subband particle filters-based tracker

No.	Step name	Computational complexity	Notes
1	*Initial background extraction step*	$O(T * N_r * N_c)$	T, N_r, and N_c are the number of averaged frames, number of row pixels and column pixels in the video frame, respectively.
2	*DWT step*	$O(N_r * N_c)$	The complexity of fast DWT is $O(N_r * N_c)$ operations [66].
3	*Object detection step*	$O(N_r * N_c)$	Subtract all pixels of current and reference frames, to obtain the difference frames.
4	*Frame denoising step*	$O\ (r^2\ log\ r)$	r is the kernel radius of the median filter.
5	*Select subbands adaptively step*	$O(N_{rs} * N_{cs})$	N_{rs}, and N_{cs} are numbers of row pixels and column pixels in the subband frame, respectively
6	*Binarize subband frames*	$O(N_{rs} * N_{cs})$	Apply a thresholding operation to all pixels.
7	*Subband particle filters step*	$O(N * N_S)$	N_S is the number of particles in a single subband particle filter.

8	*Tracker level fusion step*	$O(D_i * D_j)$	D_i and D_j are the highest two numbers of tracked objects, whose subband indices are *i* and *j*, where $1 \leq i, j \leq N$.
9	*Perform inter-frame data association step*	$\mathrm{O}(N_{ro} * N_{co} + D_p * D_c)$	N_{ro} and N_{co} are the numbers of row pixels and column pixels in the largest tracked object. D_p and D_c are the numbers of tracked objects in the previous frame current frame, respectively.

4.4 Chapter summary

We developed a robust multi-scale visual tracker that represents a captured video frame as different subbands in the wavelet domain. Our tracker then applies N independent particle filters to a small subset of these subbands, which change with each captured frame. Finally, our tracker fuses the outputs of these *N* independent particle filters to obtain at the final tracks of multiple moving objects in the video sequence. To demonstrate our tracker's robustness, we applied it to four example videos that exhibit different combinations of background motion, sudden illumination change, objects with different sizes, object shadow, and partial object camouflage. Compared to a standard full-resolution particle filter-based tracker and a single wavelet subband $(LL)_2$ based tracker, our multi-scale tracker demonstrated significantly more accurate tracking performance.

Our multi-scale tracker overcame the presence of object shadow, sudden illumination change, and background motion because of 1) applying object confirmation at the *intra-frame level*, i.e., using more than one adaptively chosen subband frames, and fusing the output position tracks obtained from multiple subband particle filters. Therefore, a truly

detected and tracked object would be verified, while a falsely detected and tracked object by a minority of the subband particle filters would be discarded, as discussed in Section 4.2.7.1. Also, 2) the frame denoising step, discussed in Section 4.2.4, decreased the impact of the presence of challenging conditions, mentioned above, in the video sequence.

We also note that the ability to handle a long shadow of an object in a full resolution frame is improved when tracking using a coarser scale frame. Also, using a denoised coarser scale frame, instead of a full resolution frame, could reduce the effect of partial object camouflage.

Chapter 5

Robust Tracking of Multiple Objects in Video by Adaptive Fusion of *N* frame Subbands Using a Cross-section Particle Filter

5.1 Introduction

In this chapter, to reduce the computational cost of our visual tracker described in Chapter 4, we develop a robust multi-scale visual tracker that adaptively fuses *N* frame subbands using a single cross-section particle filter. In this cross-section particle filter-based tracker we represent a captured video frame in the wavelet domain, and then apply a cross-section particle filter to a small subset of its wavelet subbands. The choice of this subset of wavelet subbands adaptively changes with each captured frame.

We applied our cross-section particle filter-based tracker to example videos that exhibit different combinations of challenging conditions and unexpected events. Compared to the results obtained by a standard particle filter-based tracker, our results demonstrate significantly more accurate tracking performance. Furthermore, our cross-section particle filter-based tracker requires a computational cost of approximately 50% of that required by our multi-scale tracker described in Chapter 4.

This chapter is organized as follows: Section 5.2 describes the implementation of our cross-section particle filter-based tracker. Section 5.3 presents a performance evaluation of our cross-section particle filter-based tracker. Finally, Section 5.4 provides a chapter summary.

5.2 Implementation of our cross-section particle filter-based tracker

To reduce the computational cost of our visual tracker described in Chapter 4, we developed a robust multi-scale visual tracker that used only a single particle filter that used *cross-section* estimation [117] to adaptively fuse *N* frame wavelet subbands. Figure 5.1 shows a flowchart of the implementation of our cross-section particle filter-based tracker. Most steps that were used to implement this tracker, e.g., discrete wavelet transform, current frame denoising, background extraction, intra-frame object confirmation, and inter-frame data association, were identical to the steps used in our robust multiscale tracker described in Chapter 4. The most important differences between this tracker and the multi-scale tracker described in Chapter 4 were 1) adaptive choice of the *N* frame wavelet subbands, and 2) the design and implementation of the particle filter used for the actual object tracking step. As described in Section 5.2.1, instead of choosing the three subband frames with the highest energies, we chose the three subband frames with highest energy densities. Also, instead of applying *N* independent standard particle filters to *N* frame subbands to generate *N* independent object tracks, we used a single cross-section particle filter that was *sequentially* applied to *N* frame subbands to generate *N dependent* object tracks. At every one of these *N* applications, previously generated particles, i.e., the posterior distribution resulting from a current frame subband, were used as the prior distribution for the subsequent application of this filter to the subsequent current frame subband. This would lead to a more accurate and faster estimation of object tracks from a subsequent current frame subband. The sequential application of a single cross-section particle filter to the chosen *N* frame subbands is equivalent to data fusion. The use of a

single cross-section based particle filter, instead of *N* independent standard particle filters, could result in significant reduction in the computational cost of our cross-section particle filter-based visual tracker. We will describe the design and implementation of this cross-section particle filter in Section 5.2.2

As shown in Figure 5.1, we first generated a *background frame* from the full-resolution video sequence. We then generated the subband frames by applying the *discrete wavelet transform* using a zbo6.6 wavelet to both background and current frames. Next, we obtained *difference frames* by subtracting the background subband frames from their corresponding current subbands in the current frame. Once we obtained the *difference frames*, we sequentially applied a single cross-section particle filter to three *difference frames* with the highest energy densities. Lastly, we generated the final object position tracks by fusing the three position tracks that were generated during the sequential processing of these three subbands.

In the following subsections, we describe our method for the adaptive selection of subband difference frames based on their energy densities. We also describe a *Sequential-estimation* technique for fusing *N* subbands, as well as a standard particle filter for fusing subbands that is based on a *Sequential-estimation* technique.

5.2.1 Adaptive selection of subband difference frames

Characteristics of a moving object, such as size, shape, and orientation influence the energy density of the subband *difference frames*. Thus, the adaptive selection of subband *difference frames* could be adapted to the features of moving objects.

The *difference frames*, D_t^s, were generated by subtracting the current background from the current frame, $D_t^s(i,j) = Z_t^s(i,j) - B_t^s(i,j)$, where s belongs to the set of available subband frames in wavelet level-one and level-two (shown in Figure 4.2). To approximate *difference frame* energy, we computed the l_1 norm for all of the *difference frames*. We then obtained the energy densities of the *difference frames* using

$$E(D_t^s) = \frac{1}{(I\,J)}\,|D_t^s|_1 = \frac{1}{(I\,J)}\,\sum_{i}^{I}\sum_{j}^{J}|D_t^s(i,j)| \tag{5.1}$$

Unlike our subband selection method described in Section 4.2.3, we used *difference frames'* energy densities, rather than their energies. This subband selection method would allow more selection of subbands from wavelet level-two. As the number of pixels in wavelet level-two frames is one fourth the number of pixels in wavelet level-one frames, more selection of subbands from wavelet level-two could result in significant reduction in the computational cost of our cross-section particle filter-based visual tracker.

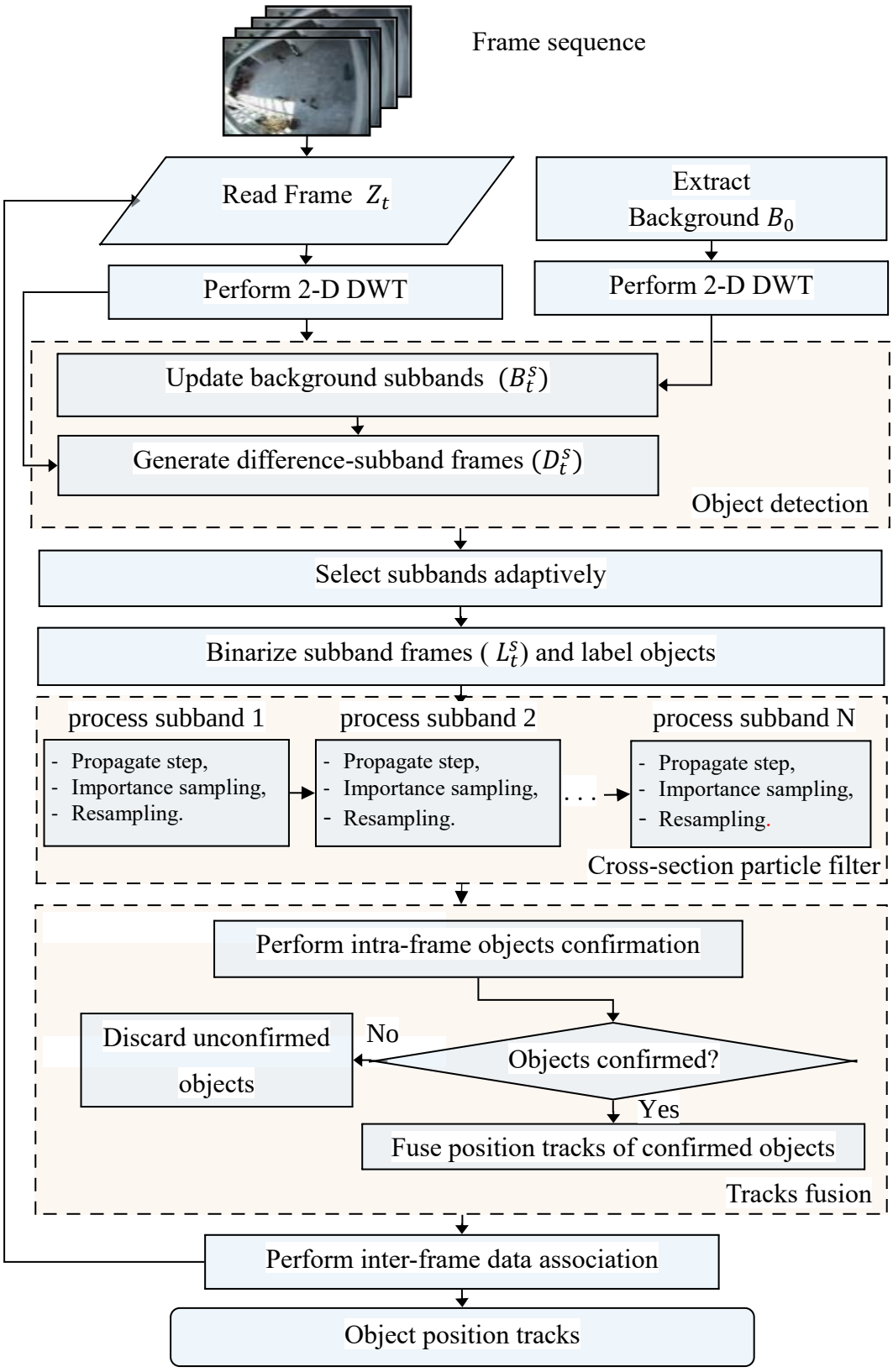


Figure 5.1. Flowchart of our cross-section particle filter-based tracker

5.2.2 Cross-section estimation technique to fuse *N* subband frames

One possible data fusion technique is *cross-section-estimation* [117]. At each current frame time point, time is frozen to sequentially process *N* subband frames.

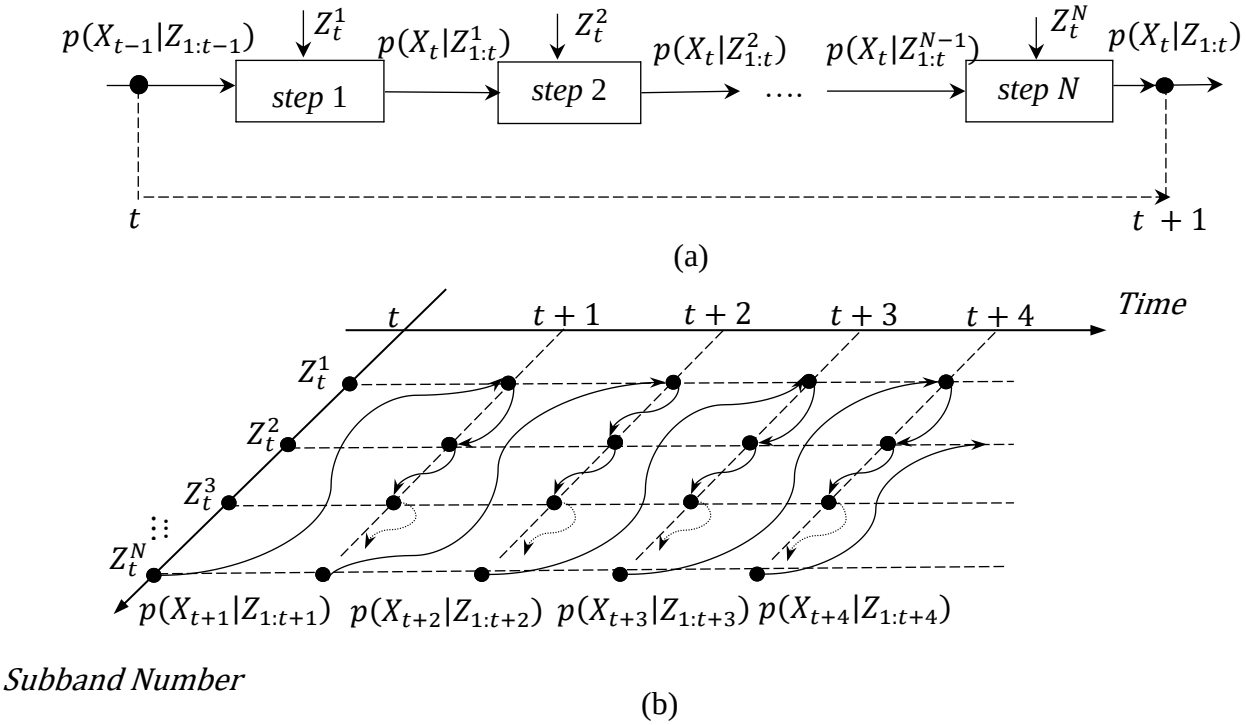


Figure 5.2. Cross-section estimation technique (a) current frame time is frozen to process *N* subband frames; (b) Cross-section estimation over the timeline

The state vector estimate, $p(X_t|Z_{1:t}^{n-1})$, that is obtained by processing the n - 1 subband frame is used as an initial esimate of the state vector when processing the subsequent *n* subband frame, thereby yielding amore accurate estimate $p(X_t|Z_{1:t}^{n})$. Figure 5.2 demonstrates this *cross-section estimation* technique; Figure 5.2 (a) shows the processing of the *N* subband frames while the current frame time is frozen at time t, and Figure 5.2(b) shows *cross-section estimation* over the timeline.

5.2.3 Particle filter for fusing *N* subbands based on cross-section estimation technique

We used a cross-section particle filter that employs the above *cross-section estimation* technique to estimate the state vector by fusing the data from the *N* chosen subband frames. Table 5.1 lists implementation details of this cross-section particle filter [112].

Table 5.1. Cross-section particle filter algorithm

With the particle set $\left\{X_{t-1}^{(i)}, w_{t-1}^{(i)}\right\}_{i=1}^{N_s}$ at the previous time $\boldsymbol{t}-\mathbf{1}$ proceed as follows at time $\boldsymbol{t}$:

Step 1: Initialization step N_s particles:

$\left\{X^{0(i)}, w^{0(i)}\right\}_{i=1}^{N_s} = \left\{X_{t-1}^{(i)}, w_{t-1}^{(i)}\right\}_{i=1}^{N_s}$

Step 2: Repeat for N subbands:

1- Propagation

- For $i = 1, \ldots, N_s$ Sample $x_t^{n(i)} \sim p_n\left(X^n \middle| X^{n-1(i)}, Z_t^n\right)$

2- Importance sampling

- For $i = 1, \ldots, N_s$ compute the importance weights $w_t^{n(i)} = p\left(Z_t^n \middle| X^{n(i)}\right)$
- Normalize the importance weights: $\sum_i \tilde{w}_t^{n(i)} = 1$

3- Resampling

- According to the normalized weights, resample with replacement N_s particles $\left\{X_t^{n(i)}\right\}_{i=1}^{N_s}$ from $\left\{\tilde{X}_t^{n(i)}\right\}_{i=1}^{N_s}$
- Set $n = n + 1$ and go to Step 2

Step 3: terminate step:

- $\left\{X_t^{(i)}, w_t^{(i)}\right\}_{i=1}^{N_s} = \left\{X^{N(i)}, w^{N(i)}\right\}_{i=1}^{N_s}$
- Set $t = t + 1$ and go to Step 1

We note that the estimate of the state vector that is obtained from processing a subband frame could be made more accurate after processing a subsequent subband [112]. Furthermore, the cross-section particle filter could save the computation cost of generating a new set of particles when a new subband frame is selected at time t and was not chosen at time $t-1$ (i.e., the initialization step of the standard particle filter). The sequential particle filter is able to do this because it uses the set of particles collected during the previous processing as initialization for the current processing step. Conversely, our former multi-scale subband particle filters tracker had to perform the initialization step each time a new subband was chosen.

Furthermore, when a new subband frame is chosen at time *t* , while its corresponding subband was also chosen at time $t-1$, our cross-section particle filter could avoid the computation cost of generating a new set of particles at time *t* by reusing the available corresponding particles from time *t* -1. This could significantly reduce the overall computation cost of our cross-section particle filter based visual tracker.

5.3 Performance evaluation of our cross-section particle filter based tracker

We evaluated the performance of our cross-section particle filter based tracker by comparing it to 1) a standard full-resolution particle filter-based tracker, and 2) our multi-scale tracker that we introduced in Chapter 4. We chose to compare our cross-section particle filter based tracker to a standard full-resolution particle filter-based tracker to demonstrate the former's superior performance and robustness in the presence of challenging video conditions and unexpected events. We also chose to compare it to our earlier multi-scale tracker to demonstrate its reduced computational cost without sacrifice

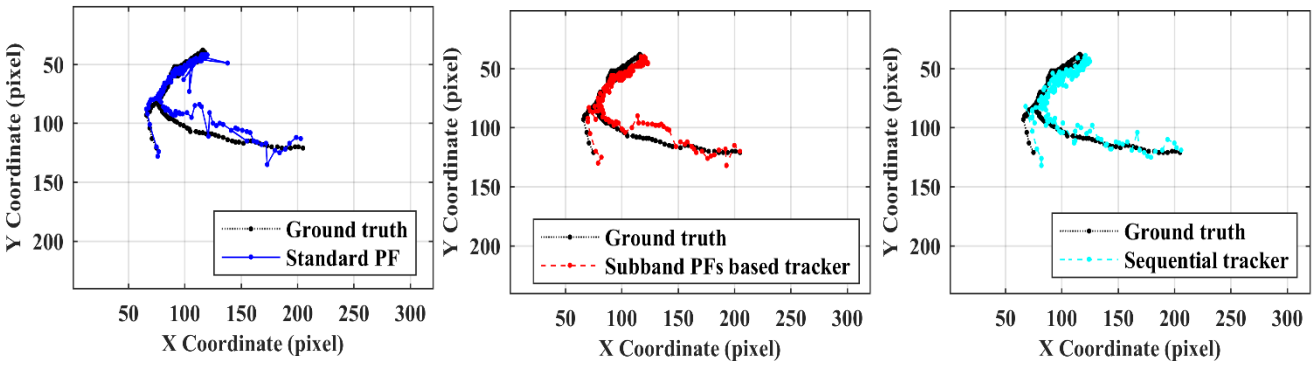
of visual tracking performance. We used four video sequences examples from VISOR and CAVIAR databases that exhibit different combinations of challenging conditions and unexpected events.

5.3.1 Example demonstrating object shadow and partial object camouflage

We applied our cross-section particle filter based tracker to a video sequence that included the presence of object shadow and partial object camouflage. The video sequence in this example is *"Intelligentroom_raw,"* and the true position track of the object, i.e., our ground truth, was also available via the VISOR database.

5.3.1.1 Comparison of resulting position tracks

We examined the resulting trajectories of our cross-section particle filter based tracker and our earlier multi-scale tracker to demonstrate that the cross-section particle filter based provides similar tracking performance while significantly reducing average frame processing times. Figure 5.3 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, our multi-scale subband particle filter tracker, and our multi-scale sequential tracker. Figure 5.3 (a), Figure 5.3 (b), and Figure 5.3 (c) show the true position tracks of the object, as well as those generated by the standard full-resolution particle filter-based tracker, our multi-scale tracker, and our cross-section particle filter based tracker, respectively. We note that the differences between the position paths generated by our cross-section particle filter based tracker and the true position paths are significantly smaller than those generated by the standard full-resolution particle filter-based tracker.



(a) (b) (c)

Figure 5.3. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter-based tracker

As shown in Table 5.2, the object in this video appeared in 217 *detection frames*, with cumulative track errors of 982 pixels, 1025 pixels, 1352 pixels using 1) standard full resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively.

We note that due to the presence of challenging conditions and unexpected events in this video sequence, e.g., object shadow; partial object camouflage; and low signal-to-noise ratio, standard full resolution particle filter-based tracker generated 90 phantom objects, while both our new trackers overcame the presence of these challenging conditions and unexpected events by not generating any phantom objects.

Table 5.2. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/300 frames)	Average position track error (pixel/*detection frame*)	Standard deviation of tracking errors	Phantom object (event/300 frames)
Full resolution PF tracker	0	4.63	4.51	90
Our multi-scale tracker	0	4.8	2.86	0
Our cross-section PF based tracker	0	6.37	2.34	0

5.3.1.2 Demonstrating challenging video conditions

Partial object camouflage: Figure 5.4 (a), Figure 5.4 (b), and Figure 5.4 (c) show the binary frames generated from the 267^{th} video frame using the full-resolution frame, subband $(LH)_2$ which is one of the three chosen subbands for this 267^{th} video frame in our implementation of our multi-scale tracker, and subband $(HL)_2$, which is one of the three chosen subbands for this 267^{th} video frame in our implementation of our cross-section particle filter based tracker, respectively.

We note that the green box in Figure 5.4 (a) highlights the division of the present object into three objects due to partial object camouflage. Figure 5.5 (a), Figure 5.5 (b), and Figure 5.5 (c) show visual tracking results, superposed onto the 267^{th} video frame,

generated by a standard full-resolution particle filter-based tracker, our multi-scale tracker, and our cross-section particle filter based tracker, respectively. We note that the standard full-resolution particle filter-based tracker generated two phantom objects due to the object division in Figure 5.4 (a), while both our multi-scale trackers overcame this presence of partial object camouflage.

(a) (b) (c)

Figure 5.4. Binary frames generated from the 267th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(HL)_1$

(a) (b) (c)

Figure 5.5. Visual tracking results for the 267th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our tracker; and (c) our cross-section particle filter based tracker

Object shadow: Figure 5.6 (a), Figure 5.6 (b), and Figure 5.6 (c) show the binary frames generated from the 240th video frame using the full-resolution frame, subband $(LL)_1$ which is one of the three chosen subbands for this 240th video frame in the implementation of our multi-scale tracker, and subband $(HL)_2$, which is one of the three chosen subbands for this 240th video frame in the implementation of our cross-section particle filter based tracker, respectively. The green box in Figure 5.6 (a) highlights an artifact due to the presence of object shadow.

(a) (b) (c)

Figure 5.6. Binary frames generated from the 240th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(HL)_2$

(a) (b) (c)

Figure 5.7. Visual tracking results for the 240th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker

Figure 5.7 (a), Figure 5.7 (b), and Figure 5.7 (c) show the visual tracking results, superposed on the 240th video frame, obtained by the standard full-resolution particle filter-based tracker, our multi-scale tracker, and our cross-section particle filter based tracker, respectively. We note that the standard full-resolution particle filter-based tracker generated a phantom object due to the artifact in Figure 5.6 (a), while our multi-scale trackers overcame the presence of object shadow.

5.3.2 Example demonstrating partial object camouflage and background motion

For this example, we used the "*OneLeaveShopReenter2front"* video sequence from the CAVIAR database. In this video sequence, two people walk past the front of a store, while another person exits the store and then re-enters it.

5.3.2.1 Comparison of resulting position tracks

We examined the resulting object trajectories of our cross-section particle filter based tracker, and our multi-scale tracker to confirm that our cross-section particle filter based tracker provides comparable tracking performance, but with significantly improved average frame processing times. We note that the true position tracks of the moving objects, i.e., our ground truth, in the *"OneLeaveShopReenter2front"* example were available from the CAVIAR database.

Figure 5.8 shows the position tracks of objects obtained using our multi-scale tracker and our cross-section particle filter based tracker. Figure 5.8 (a) - (c) and Figure 5.8 (d) - (f) show the actual position tracks of the three objects, superposed on position tracks

obtained by our multi-scale tracker, and our cross-section particle filter based tracker, respectively.

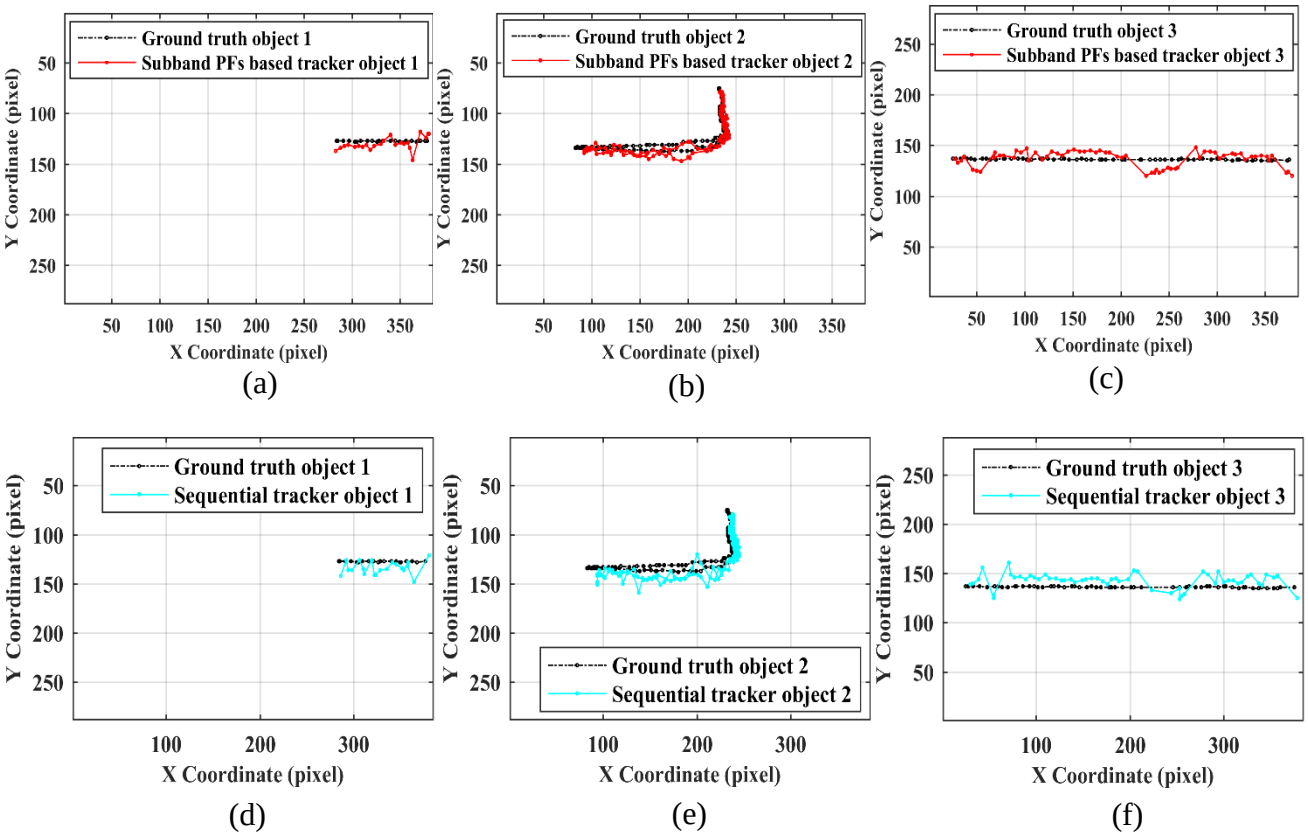


Figure 5.8. Position tracks of true objects (a) - (f) in the *"OneLeaveShopReenter2front"* video using our multi-scale tracker (upper row), and our cross-section particle filter based tracker (lower row)

Table 5.3 shows that object 1 in this video appeared in 57 *detection frames*, with cumulative track errors of 364 pixels, 316 pixels, 505 pixels using 1) standard full-resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively. Object 2 in this video appeared in 470 *detection frames*, with cumulative track errors of 1874 pixels, 797 pixels, 3706 pixels using 1) standard full

resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively. Object 3 in this video appeared in 120 *detection frames*, with cumulative track errors of 1583 pixels, 797 pixels, 1211 pixels using 1) standard full resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively. These values are a solid demonstration of the superior performance and robustness of our multiscale trackers compared to the other two trackers.

Table 5.3. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker	Missed object (event/558 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/558 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	55	6.38	4	13.19	3.68	3.12	5.60	469
N subband PFs tracker	23	5.56	4.57	6.6	3.29	2.36	2.75	0
Cross-section PF tracker	30	8.87	7.8	10	4.12	4.29	3.95	0

5.3.2.2 Demonstrating challenging video conditions

The challenging conditions present in this video sequence were partial object camouflage, object shadow, and background motion.

Partial object camouflage: Figure 5.9 (a), Figure 5.9 (b), and Figure 5.9 (c) show the binary frames generated from the 88th video frame using the full-resolution frame, subband $(LL)_1$ which is one of the three chosen subbands for this 88th video frame in our implementation of our multi-scale tracker, and subband $(HL)_2$, which is one of the three chosen subbands for this 88th video frame in our implementation of our cross-section particle filter based tracker, respectively.

We note that the green box in Figure 5.9 (a) highlights the division of an object into two objects due to partial object camouflage. Figure 5.10 (a), Figure 5.10 (b), and Figure 5.10 (c) show the visual tracking results, superposed on the 88th video frame, generated by the standard full-resolution particle filter-based tracker, our multi-scale tracker, and our cross-section particle filter based tracker, respectively.

(a) (b) (c)

Figure 5.9. Binary frames generated from the 88th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(HL)_2$

(a) (b) (c)

Figure 5.10. Visual tracking results for the 88th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale filter-based tracker; and (c) our cross-section particle filter based tracker

We note that the standard full-resolution particle filter-based tracker produced a phantom object due to the object division in Figure 5.10 (a), while our multi-scale trackers overcame the presence of partial object camouflage.

Object shadow: Figure 5.11 (a), Figure 5.11 (b), and Figure 5.11 (c) show the binary frames generated from the 116th video frame using the full-resolution frame, subband $(LL)_1$ which is one of the three chosen subbands for this 116th video frame in our implementation of our multi-scale tracker, and subband $(HL)_2$, which is one of the three chosen subbands for this 116th video frame in our implementation of our cross-section particle filter based tracker, respectively.

(a) (b) (c)

Figure 5.11. Binary frames generated from the 116th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(HL)_2$

(a) (b) (c)

Figure 5.12. Visual tracking results for the 116th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker

We note that the green box in Figure 5.12 (a) highlights an artifact due to the presence of object shadow. Figure 5.12 (a), Figure 5.12 (b), and Figure 5.12 (c) show the visual tracking results, superposed on the 116th video frame, generated by a standard full-resolution particle filter-based tracker, our multi-scale tracker, and our cross-section particle filter based tracker, respectively. We also note that the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 5.11 (a).

Background motion: Figure 5.13 (a), Figure 5.13 (b), and Figure 5.13 (c) show the binary frames generated from the 138th video frame using the full-resolution frame, subband $(LL)_1$ which is one of the three chosen subbands for this 138th video frame in our implementation of our multi-scale tracker, and subband $(LL)_2$, which is one of the three chosen subbands for this 138th video frame in our implementation of our cross-section particle filter based tracker, respectively.

We note that the green box in Figure 5.13 (a) highlights an artifact due to the presence of background motion. Figure 5.14 (a), Figure 5.14 (b), and Figure 5.14 (c) show the visual tracking results, superposed onto the 138th video frame, generated by the standard full-resolution particle filter-based tracker, our multi-scale tracker, and our cross-section particle filter based tracker, respectively.

(a) (b) (c)

Figure 5.13. Binary frames generated from the 138th frame using: (a) the full-resolution frame; (b) subband $(LL)_1$; (c) subband $(LL)_2$

(a) (b) (c)

Figure 5.14. Visual tracking results for the 138th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker

We also note that the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 5.13 (a), while our multiscale trackers overcame the presence of background motion.

5.3.3 Example demonstrating illumination change, objects of different sizes, and partial object camouflage

In this example, we used the *"Meet_WalkTogether2"* video sequence from the CAVIAR database. In this video sequence, two people meet and then walk together.

5.3.3.1 Comparison of resulting position tracks

We examined the resulting object trajectories of our cross-section particle filter based tracker, and our multi-scale tracker to confirm that our cross-section particle filter based tracker provides comparable tracking performance, but with significantly improved average frame processing times. We note that the true position tracks of the moving

objects, i.e., our ground truth, in the *"Meet_WalkTogether2"* example were available from the CAVIAR database.

Figure 5.15 shows the position tracks of objects obtained using our multi-scale tracker and our cross-section particle filter-based tracker. Figure 5.15 (a) – (c) and Figure 5.15 (d) - (f) show the actual position tracks of the three objects, in addition to ones generated by the multi-scale tracker, and our cross-section particle-filter based tracker, respectively.

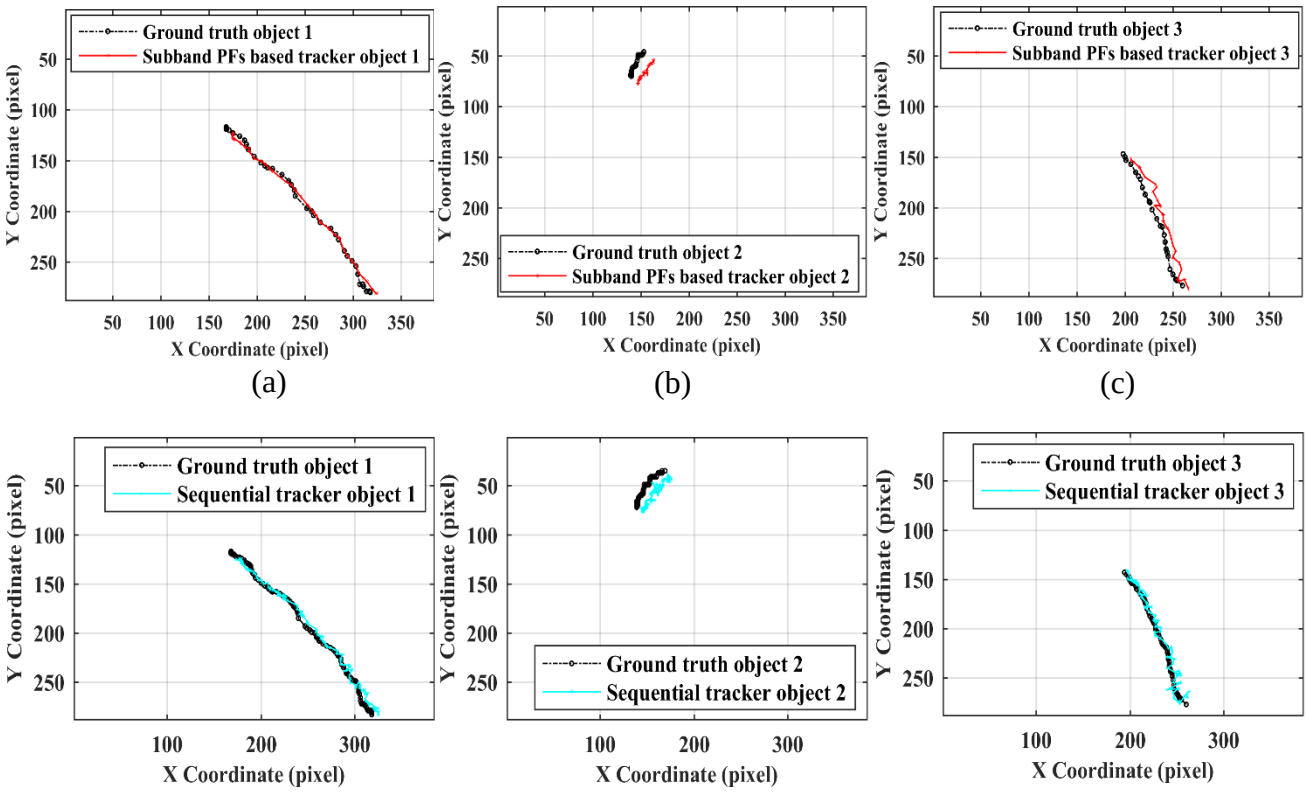


Figure 5.15. Position tracks of true objects (a) - (f) in the *"Meet_WalkTogether2"* video using our multi-scale tracker (upper row), and our cross-section particle filter based tracker (lower row)

As shown in Table 5.4 , object 1 in this video appeared in 109 *detection frames*, with cumulative track errors of 721 pixels, 547 pixels, 1146 pixels using 1) standard full resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively. Object 2 in this video appeared in 8 *detection frames*, with cumulative track errors of 72 pixels, 80 pixels, 72 pixels using 1) standard full resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively. Object 3 in this video appeared in 59 *detection frames*, with cumulative track errors of 703 pixels, 641 pixels, 304 using 1) standard full resolution particle filter, 2) our multi-scale tracker, and 3) our cross-section particle filter based tracker, respectively.

Table 5.4. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/827 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/827 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	81	6.6	9.04	11.92	7.9	3.91	3.19	122
N subband PFs tracker	45	5.2	10.05	10.8	3.52	0.69	3.42	0
Cross-section PF tracker	55	10.5	9.12	5.16	3.31	1.37	2.76	0

5.3.3.2 Demonstrating challenging video conditions

The challenging conditions present in this video sequence were sudden illumination change, the presence of objects of different sizes and partial object camouflage:

The presence of sudden illumination change: Figure 5.16 (a), Figure 5.16 (b), and Figure 5.16 (c) show the binary frames generated from the 67^{th} video frame using the full-resolution frame, subband $(LL)_1$ which is one of the three chosen subbands for this 67^{th} video frame in our implementation of our multi-scale tracker, and subband $(LL)_2$, which is one of the three chosen subbands for this 67^{th} video frame in our implementation of our cross-section particle filter based tracker, respectively. We note that the green box in Figure 5.16 (a) highlights artifacts due to a sudden illumination change in this video frame. Figure 5.17 (a), Figure 5.17 (b), and Figure 5.17 (c) show the visual tracking results, superposed on the 67^{th} video frame, generated by a standard full-resolution particle filter-based tracker and our multi-scale trackers, respectively.

(a) (b) (c)

Figure 5.16. Binary frames generated from the 67^{th} frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) subband $(LH)_2$

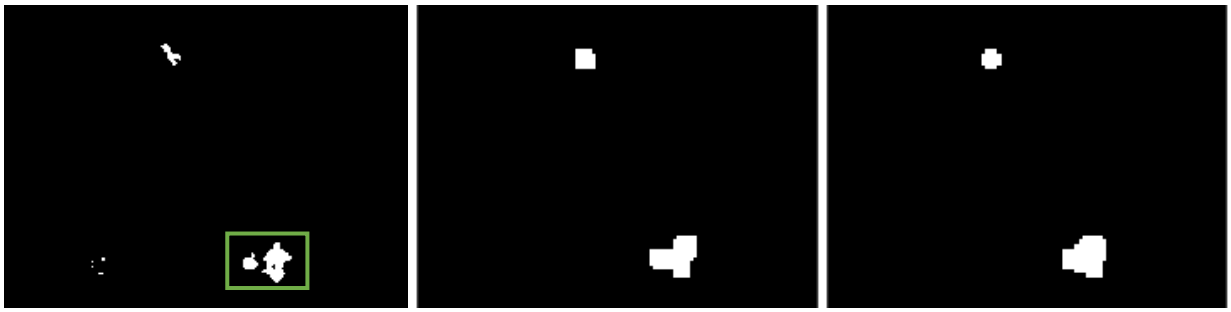
(a) (b) (c)

Figure 5.17. Visual tracking results for the 67th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker

We also note that the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 5.16 (a), while our multi-scale trackers overcame the effect of sudden illumination change in this 67th video frame.

The presence of objects of different sizes and partial object camouflage: Figure 5.18 (a), Figure 5.18 (b), and Figure 5.18 (c) show the binary frames generated from the 202nd video frame using the full-resolution frame, subband $(LH)_2$ which is one of the three chosen subbands for this 202nd video frame in our implementation of our multi-scale tracker, and subband $(HL)_2$, which is one of the three chosen subbands for this 202nd video frame in our implementation of our cross-section particle filter based tracker, respectively. We note that 1) the object sizes in Figure 5.18 (b), and Figure 5.18 (c) are closer to each other than the object sizes in Figure 5.18 (a); and 2) the green box in Figure 5.18 (a) highlights the division of an object into two objects due to the presence of partial object camouflage. Figure 5.19 (a), Figure 5.19 (b), and Figure 5.19 (c) show visual tracking results, superposed on the 202nd video frame, generated by the standard full-resolution

particle filter-based tracker and our multi-scale trackers, respectively. We also note that, due to the presence of a large object, the standard full-resolution particle filter-based tracker failed to track the smaller object. Also, the standard full-resolution particle filter-based tracker generated a phantom object due to the presence of the object division in Figure 5.18 (a).



(a) (b) (c)

Figure 5.18. Binary frames generated from the 202nd frame using: (a) the full-resolution frame; (b) subband $(LH)_2$; (c) subband $(HL)_2$

(a) (b) (c)

Figure 5.19. Visual tracking results for the 202nd video frame using: (a) the standard full-resolution particle filter-based tracker; (b) our multi-scale tracker; and (c) our cross-section particle filter based tracker

5.3.4 Example demonstrating presence of objects with different sizes and partial object camouflage

The The video sequence in this example, "ATCS" is from the Visor database (288 X 384 pixels, 30 fps, 1313 frames). This video sequence shows three moving people.

5.3.4.1 Comparison of resulting position tracks

Figure 5.21 shows the position tracks of objects obtained using our multi-scale subband particle filters tracker, and our multi-scale cross section filter tracker. Figure 5.21 (a) - (c), and Figure 5.21 (d) - (f) show the true position tracks of the three objects, as well as those generated by our multi-scale subband particle filters tracker, and our multi-scale cross section filter tracker, respectively. We note that the differences between the position paths generated by our multi-scale trackers and the true position paths are almost the same. Figure 5.20 shows the visual tracking results for a sample of four video frames using our multi-scale tracker. We note that our multi-scale trackers generated no phantom objects. This is a further demonstration of our multi-scale tracker's robustness.

Table 5.5 shows that object 1 video appeared in 384 *detection frames*, with cumulative track errors of 2175 pixels, 3891 pixels, 4428 pixels using 1) standard full resolution particle filter, 2) our subband particle filters tracker, and 3) our multi-scale cross section filter tracker, respectively. Object 2 in this video appeared in 240 *detection frames*, with cumulative track errors of 1690 pixels, 2886 pixels, 3119 pixels using 1) standard full resolution particle filter, 2) our subband particle filters tracker, and 3) our multi-scale cross section filter tracker, respectively. Object 3 in this video appeared in 294 *detection frames*,

with cumulative track errors of 1655 pixels, 2580 pixels, 3139 pixels 1) standard full resolution particle filter, 2) our subband particle filters tracker, and 3) our multi-scale cross section filter tracker, respectively. These values are a solid demonstration of the superior performance and robustness of our multiscale tracker compared to the other two trackers. We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than those generated by the standard full-resolution particle filter-based tracker and the LL-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 83, 34, and 15 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the number of times that a real object failed to be tracked was 83, 34, and 7 for 1) standard full resolution particle filter,

2) our subband particle filters tracker, and 3) our multi-scale cross section filter tracker, respectively.

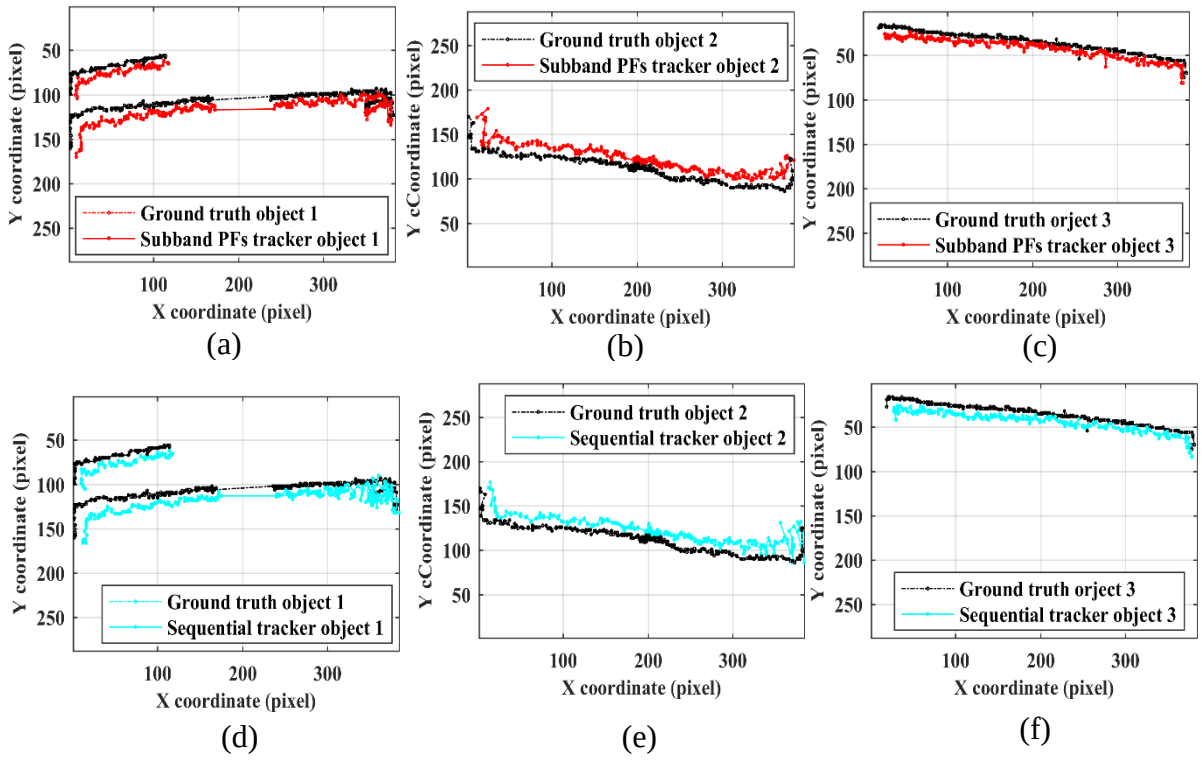


Figure 5.21. Position tracks of true objects (a) - (f) in the "*ATCS*" video using our multi-scale tracker (upper row), and our cross-section particle filter based tracker (lower row)

Figure 5.20. Visual tracking results for four video frames using our multi-scale tracker

Table 5.5. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/1313 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/1313 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	83	5.65	7.04	5.62	2.95	3.42	4.49	31
N subband PFs tracker	15	10.13	12.02	8.59	4.1	4.08	3.30	0
Cross-section PF tracker	7	11.53	12.9	10.67	4.45	5.15	3.85	0

5.3.5 Average frame processing times

In addition to demonstrating our proposed sequential multi-scale tracker's robustness, we also confirmed that it incurs less computation cost than our previous subband particle filter-based tracker. We examined the average frame processing times for both trackers, and we obtained our results using Matlab R2016 running on a 2.8 GHz Intel R Core™ i7 with 4 GB RAM. In Table 5.6, we compare the average frame processing times for our previous *N* independent subband particle filter-based tracker and our proposed sequential particle filter-based tracker. We note that our proposed tracker had average frame processing times that were approximately 50% faster than the average times produced by our previous tracker.

Table 5.6. Average frame computation times using our *N* subband particle filter-based tracker and our proposed sequential particle filter-based tracker

Video sequences	Data-base	*N* subband particle filter-based tracker (sec/ frame)		Sequential particle filter-based tracker (sec/ frame)	
		3000 Particles	5000 Particles	3000 Particles	5000 Particles
"Intelligentroom_raw"	Visor	0.0379	0.0436	0.0192	0.0228
"OneLeaveShopReenter2front"	Caviar	0.0582	0.0694	0.0292	0.0314
"Meet_WalkTogether2"	Caviar	0.0556	0.0636	0.0274	0.0303

5.4 Chapter summary

In this chapter, to reduce the computational cost of our visual tracker described in Chapter 4, we developed arobust multi-scale visual tracker that adaptively fused *N* frame subbands using a single cross-section particle filter. In this cross-section particle filter-based tracker we represented a captured video frame in the wavelet domain, and then applied a cross-section particle filter to a small subset of its wavelet subbands. The choice of this subset of wavelet subbands adaptively changes with each captured frame.

We applied our cross-section particle filter-based tracker to example videos that exhibit different combinations of challenging conditions and unexpected events. Compared to the results obtained by a standard particle filter-based tracker, our results demonstrate significantly more accurate tracking performance. Furthermore, our cross-section particle

filter-based tracker required a computational cost of approximately 50% of that required by our multi-scale tracker described in Chapter 4.

Chapter 6

Dual-tree complex wavelet transform for robust visual tracking

6.1 Introduction

In this chapter, we present a robust multi-scale visual tracker that represents a captured video frame as different subbands of the *dual-tree complex wavelet transform* (DW-CWT). Once this has been done, the tracker then applies N independent particle filters to a small subset of these subbands. The choice of this subset of wavelet subbands changes adaptively with each captured frame. Finally, the tracker fuses the outputs of these N independent particle filters, i.e., tracker-level fusion, to obtain the final position tracks of multiple moving objects in the video sequence.

Real-valued wavelet transforms have been previously used for visual tracking, but most suffer from shift variance and low directional selectivity. Therefore, we used DT-CWT to avoid such shortcomings. To demonstrate our visual tracker's robustness, we applied it to videos with challenging visual conditions. When compared to the performance of a standard full-resolution particle filter-based tracker, and a single wavelet subband $(LL)_2$ based tracker, our DT-CWT multi-scale tracker achieved significantly more accurate tracking results.

This chapter is organized as follows: Section 6.2 introduces the Dual-Tree Complex Wavelet. Section 6.3 presents the performance evaluation of our DT-CWT tracker. Finally, Section 6.4 provides a summary of this chapter.

6.2 Dual-Tree Complex Wavelets

The dual-tree complex wavelet transform was introduced by Nick Kingsbury [118] as an enhancement of the real-valued wavelet transform [119]. Compared to the real-valued wavelet transform, the DT-CWT has additional superior features, including:

1. More directional selectivity features. The DW-CWT can detect the edges in an image along six directions at different resolution scales, compared to the DWT that detects edges at the vertical, horizontal and diagonal orientations only.
2. Shift invariance features. A shift in a signal does not produce a shift in the coefficients of the subbands. This is achieved with a limited redundancy factor of only 4 for 2-D images [119].

To address the above limitations associated with using a standard particle filter for visual tracking, we used the *dual-tree complex wavelet transform* to represent captured video frames, as it has several advantages for visual tracking, including:

1. A wavelet transform. The dual-tree complex wavelet transform is suitable for tracking objects with different sizes or contrast levels that may be present in the same video frame, as it produces subband frames with different resolutions (scales). Subband frames with a coarse resolution (large scale) are more suitable for tracking large objects or objects with high contrast, while subband frames with a fine resolution (small scale) are more appropriate for tracking small objects and/or objects with a low contrast [61].
2. In visual tracking, different types of object motion, such as translational and

rotational motion, are detected across subsequent frames. Therefore, representing these motion translations using a shift-invariant transform like dual-tree complex wavelet transform could produce better visual tracking results [120].

3. Dual-tree complex wavelet transform is a natural edge detector that can detect the boundaries of objects in various directions and is sensitive to edges along six different directions ($\pm 15^0,\ \pm 45^0, and\ \pm 75^0$).
4. Denoising a video sequence using wavelets is relatively easy. Typically, denoising is performed by setting small wavelet coefficients to zero. The use of a shift-invariant wavelet transform like the dual-tree complex wavelet transform will typically result in better denoising performance than a shift-variant wavelet transform [65].

6.3 Performance evaluation of our robust DT-CWT based tracker

In the following examples, we show that our DT-CWT based tracker overcame the presence of challenging conditions in three video sequences. Moreover, we show that our tracker demonstrated better tracking performance compared to a typical visual tracker using a standard full-resolution particle filter-based, and a single wavelet subband $(LL)_2$ based tracker.

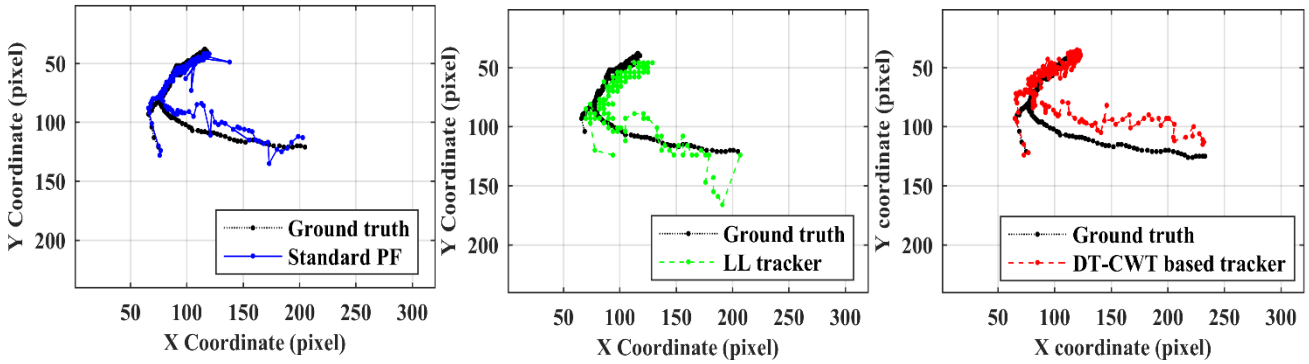
6.3.1 Example demonstrating object shadow, and partial object camouflage

In this example, we used the *"Intelligentroom_raw"* video sequence from the Visor database. In this video sequence, a man walking around a conference room. We note that the challenging conditions present in this video sequence were: object shadow and partial object camouflage.

6.3.1.1 Comparison of resulting position tracks

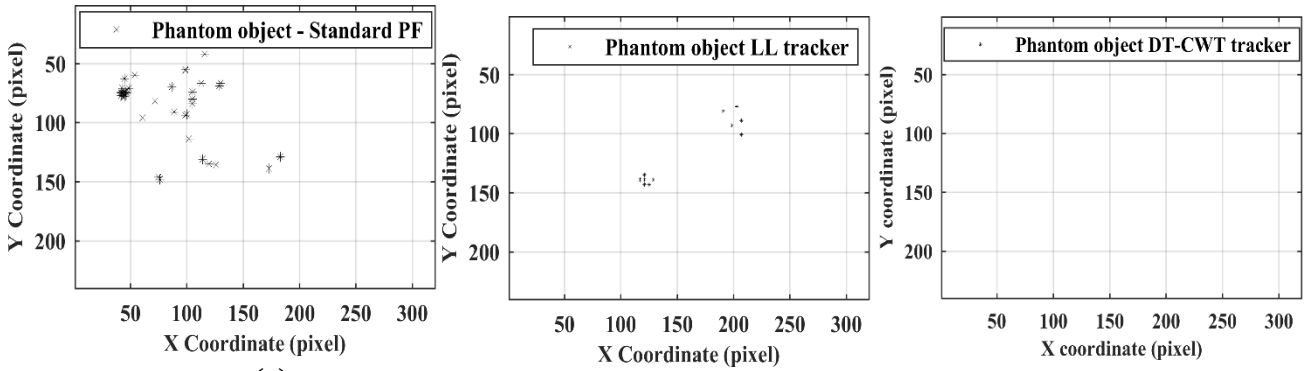
Figure 6.1 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ tracker, and our DT-CWT based tracker. Figure 6.1 (a), Figure 6.1 (b), and Figure 6.1 (c) show the true position tracks of the object, as well as those generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our DT-CWT based tracker, respectively.

Figure 6.2 shows the position tracks of phantom objects generated by the standard full-resolution particle filter-based tracker and the single wavelet subband $(LL)_2$ based tracker. These phantom objects may have appeared due to the presence of the object's shadow or partial object camouflage. We note that our DT-CWT based tracker generated no phantom objects, while the standard full-resolution particle filter-based tracker and the single wavelet subband $(LL)_2$ based tracker generated many.



(a) (b) (c)

Figure 6.1. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) a single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker



(a) (b) (c)

Figure 6.2. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

To quantitatively compare the performance of our visual tracker considered here, we compared it to a standard full-resolution particle filter-based tracker, and a single wavelet subband $(LL)_2$ based tracker.

Table 6.1 shows that the object in this video appeared in 214 *detection frames*, with cumulative track errors of 987 pixels, 2032 pixels, 2032 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale DT-CWT based tracker, respectively.

We note that due to the presence of challenging conditions and unexpected events in this video sequences, .e.g., object shadow; partial object camouflage; and low signal-to-noise ratio, standard full resolution particle filter-based tracker generated 90 phantom objects, also the single wavelet subband $(LL)_2$ based tracker generated 11 phantom objects, while our multi-scale DT-CWT based tracker overcame the presence of these challenging conditions and unexpected events and generated no phantom objects.

Table 6.1. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/300 frames)	Average position track error (pixel/*detection frame*)	Standard deviation of tracking errors	Phantom object (event/300 frames)
Full resolution PF tracker	0	4.635	4.5	90
$(LL)_2$ subband tracker	2	9.5	5.8	11
DT-CWT based tracker	0	9.5	7.0	0

These values are a solid demonstration of the superior performance and robustness of our multi-scale tracker compared to the other two trackers.

6.3.1.2 Demonstrating challenging video conditions

Object shadow: Figure 6.3 (a), Figure 6.3 (b), and Figure 6.3 (c) show the binary frames generated from the 100th video frame using the full-resolution frame, subband $(LL)_2$, and one of the chosen subbands in our multi-scale DT-CWT based tracker, respectively. We note that the green box in Figure 6.3 (a) highlights an artifact due to the presence of object shadow. Figure 6.4 (a), Figure 6.4 (b), and Figure 6.4 (c) show visual tracking results, superposed onto the 100th video frame, generated by a standard full-resolution particle filter-based tracker, a single wavelet subband $(LL)_2$ based tracker, and our DT-CWT based tracker, respectively. We note that our visual tracker was able to overcome the presence of object shadow in this video frame.

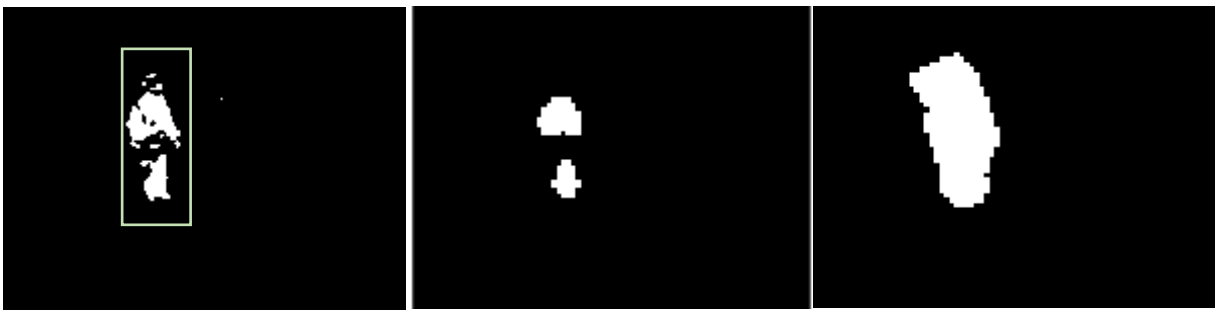
(a) (b) (c)

Figure 6.3. Binary frames generated from the 100th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) one of the chosen subbands in our multi-scale DT-CWT based tracker

(a) (b) (c)

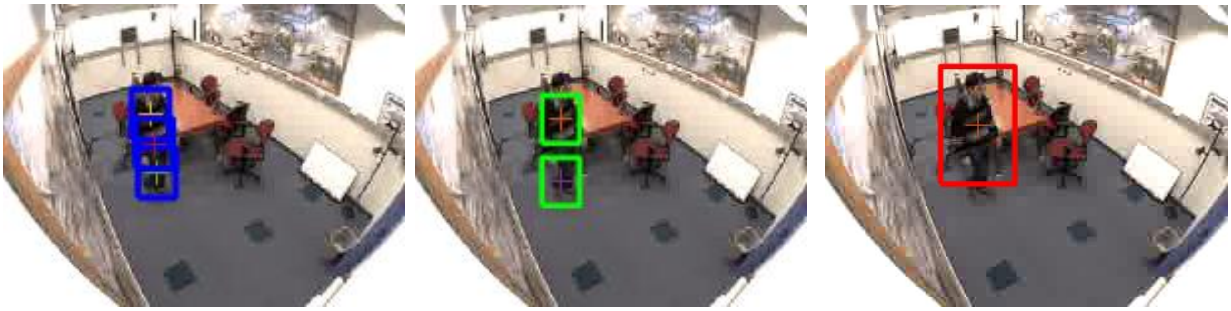
Figure 6.4. Visual tracking results for the 100th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

Partial object camouflage: Figure 6.5 (a), Figure 6.5 (b), and Figure 6.5 (c) show the binary frames generated from the 267th video frame using the full-resolution frame, subband $(LL)_2$, and one of the chosen subbands in our multi-scale DT-CWT based tracker, respectively. We note that the green box in Figure 6.5 (a) highlights an artifact due to the presence of object shadow.



(a) (b) (c)

Figure 6.5. Binary frames generated from the 267th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) one of the chosen subbands in our multi-scale DT-CWT based tracker



(a) (b) (c)

Figure 6.6. Visual tracking results for the 267th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

Figure 6.6 (a) and Figure 6.6 (b) the visual tracking results, superposed on the 267th video frame, generated by a standard full-resolution particle filter-based tracker and our visual tracker, respectively. We note from Figure 6.6 (c) that our multi-scale tracker overcame the presence of partial object camouflage in this video frame.

6.3.2 Example demonstrating object shadow, and partial object camouflage

In this example, we used the *"OneLeaveShopReenter2front"* video sequence from the CAVIAR database. In this video sequence, two people walk past the front of a store, while another person exits the store and then re-enters it. We note that the challenging conditions present in this video sequence were the presence of object shadow and partial object camouflage.

6.3.2.1 Comparison of resulting position tracks

Figure 6.7 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker. Figure 6.7 (a)-(c), Figure 6.7 (d)-(f), and Figure 6.7 (g)-(j)

show the true position tracks of the three objects, as well as those generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker, respectively.

We note that the differences between the position paths generated by our DT-CWT based tracker and the true position paths are significantly smaller than the differences generated by the standard full-resolution particle filter-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 55, 80, and 33 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker, respectively.

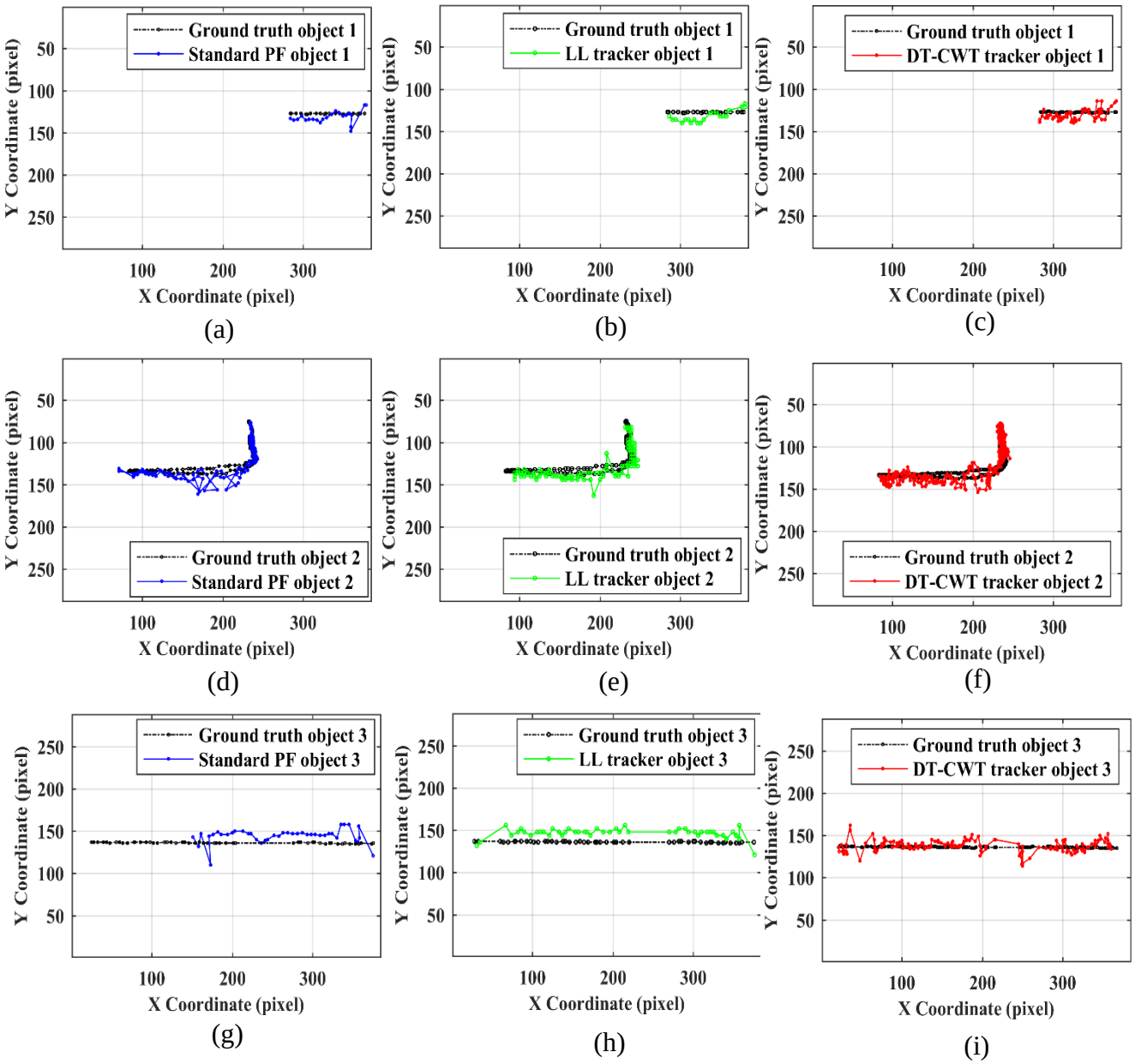
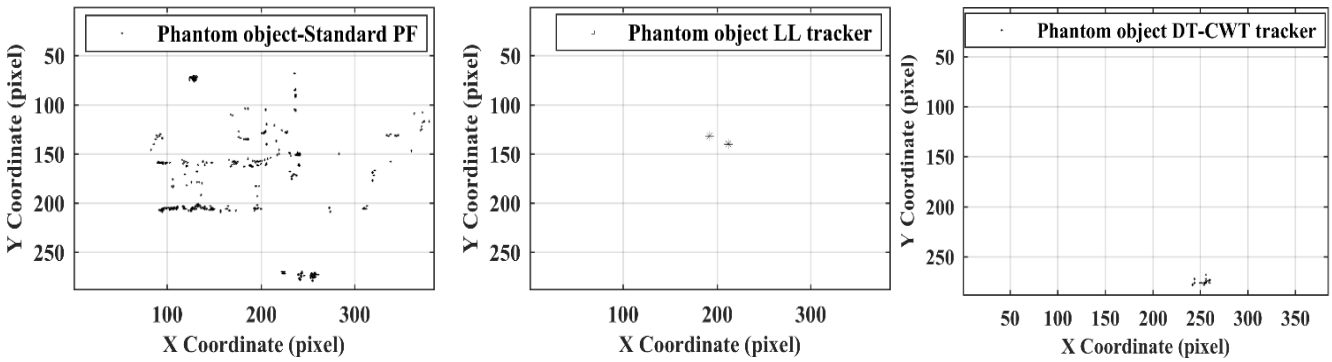


Figure 6.7. Position tracks of true objects (a) - (i) in the *''OneLeaveShopReenter2front''* video using a standard full-resolution particle filter-based tracker (right column), an LL-based tracker (middle column), and our multi-scale DT-CWT based tracker (left column)

Figure 6.8 shows the position tracks of phantom objects generated by the standard full-resolution particle filter-based tracker. These phantom objects may have appeared due to the presence of background motion, object shadows, or partial object camouflage.



(a) (b) (c)

Figure 6.8. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker; and (c) our multi-scale DT-CWT based tracker

As a result of detecting the edges in video frames along six directions in our DT-CWT based tracker, it generated phantom objects due to the presence of background motion. We note that the generated phantom objects were 479 phantom objects, 2 phantom objects and phantom 28 objects using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale DT-CWT based tracker, respectively.

As shown in Table 6.2, object 1 in this video appeared in 54 detection frames, with cumulative track errors of 303 pixels, 439 pixels, 362 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale tracker, respectively. Object 2 in this video appeared in 456 *detection frames*, with cumulative track errors of 1820 pixels, 3296 pixels, 2829 pixels using 1) standard full resolution particle

filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale DT-CWT based tracker, respectively. Object 3 in this video appeared in 116 *detection frames*, with cumulative track errors of 1518 pixels, 1442 pixels, 632 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our DT-CWT based tracker, respectively.

Table 6.2. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/558 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/558 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	55	5.62	3.99	13.08	2.91	3.09	5.45	469
$(LL)_2$ subband tracker	80	8.13	7.23	12.43	3.99	3.01	3.43	2
DT-CWT based tracker	33	6.7	6.2	5.45	3.54	3.70	3.16	28

6.3.2.2 Demonstrating challenging video conditions

Object shadow: Figure 6.9 (a), Figure 6.9 (b), and Figure 6.9 (c) show the binary frames generated from the 305^{th} video frame using the full-resolution frame, subband $(LL)_2$, and one of the chosen subbands using our DT-CWT based tracker, respectively.

We note that the green box in Figure 6.9(a) highlights an artifact due to the presence of object shadow. Figure 6.10(a), Figure 6.10(b), and Figure 6.10(c) the visual tracking

results, superposed on the 305th video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our DT-CWT based visual tracker, respectively. We note that our multi-scale tracker overcame the presence of object shadow in this video frame.

(a) (b) (c)

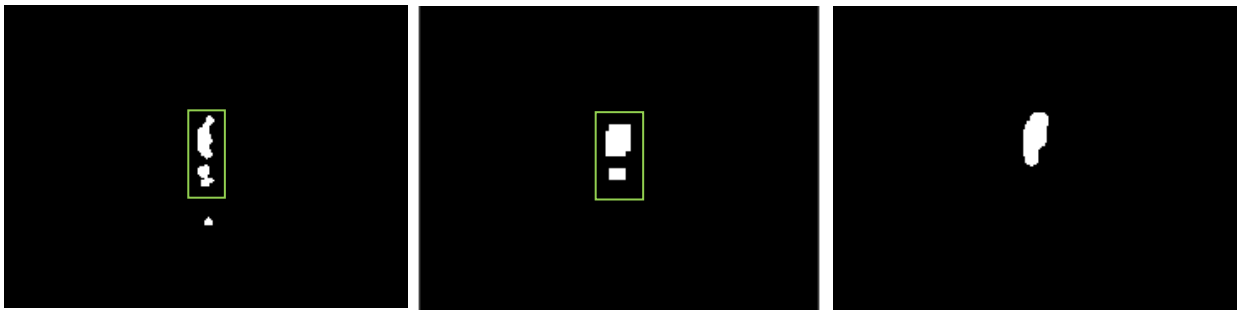
Figure 6.9. Binary frames generated from the 305th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) one of the chosen subbands in our multi-scale DT-CWT based tracker

(a) (b) (c)

Figure 6.10. Visual tracking results for the 305th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

Partial object camouflage: Figure 6.11 (a), Figure 6.11 (b), and Figure 6.11 (c) show the binary frames generated from the 427th video frame using the full-resolution

frame, subband $(LL)_2$, and, one of the chosen subbands in our multi-scale DT-CWT based tracker, respectively. We note that the green box in both Figure 6.11 (a) and Figure 6.11 (b) highlights the division of an object into two objects due to the presence of partial object camouflage.



(a) (b) (c)

Figure 6.11. Binary frames generated from the 427th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) a chosen subband by multi-scale DT-CWT based tracker

(a) (b) (c)

Figure 6.12. Visual tracking results for the 427th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

Figure 6.12 (a), Figure 6.12 (b), and Figure 6.12 (c) show the visual tracking results, superposed onto the 427th video frame, generated by the standard full-resolution particle

filter-based tracker and our multi-scale DT-CWT based tracker, respectively. We note that our visual tracker overcame the presence of partial object camouflage in this video frame.

6.3.3 Example demonstrating a change in illumination and objects of different sizes

In the second example, we used the *"Meet_WalkTogether2"* video sequence from the CAVIAR database. In this sequence, two people meet and walk together. The challenging conditions present in this video sequence are a change in illumination and the presence of objects of different sizes.

6.3.3.1 Comparison of resulting position tracks

Figure 6.13 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband $(LL)_2$ based tracker, and our DT-CWT based visual tracker. Figure 6.13 (a) - (c), Figure 6.13 (d) - (f), and Figure 6.13 (g) - (i) show the true position tracks of the three objects, as well as those generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker, respectively. We note that the differences between the position paths generated by our multi-scale DT-CWT based tracker and the true position paths are significantly smaller than the differences generated by the standard full-resolution particle filter-based tracker and single wavelet subband $(LL)_2$ tracker.

Figure 6.14 shows the position tracks of phantom objects generated by the standard full-resolution particle filter-based tracker. These phantom objects may have appeared due to the presence of background motion, object shadows, or partial object camouflage. We

note that our Figure 6.14 generated one phantom object, while the standard full-resolution particle filter-based tracker generated many. This is a further demonstration of our multi-scale tracker's robustness.

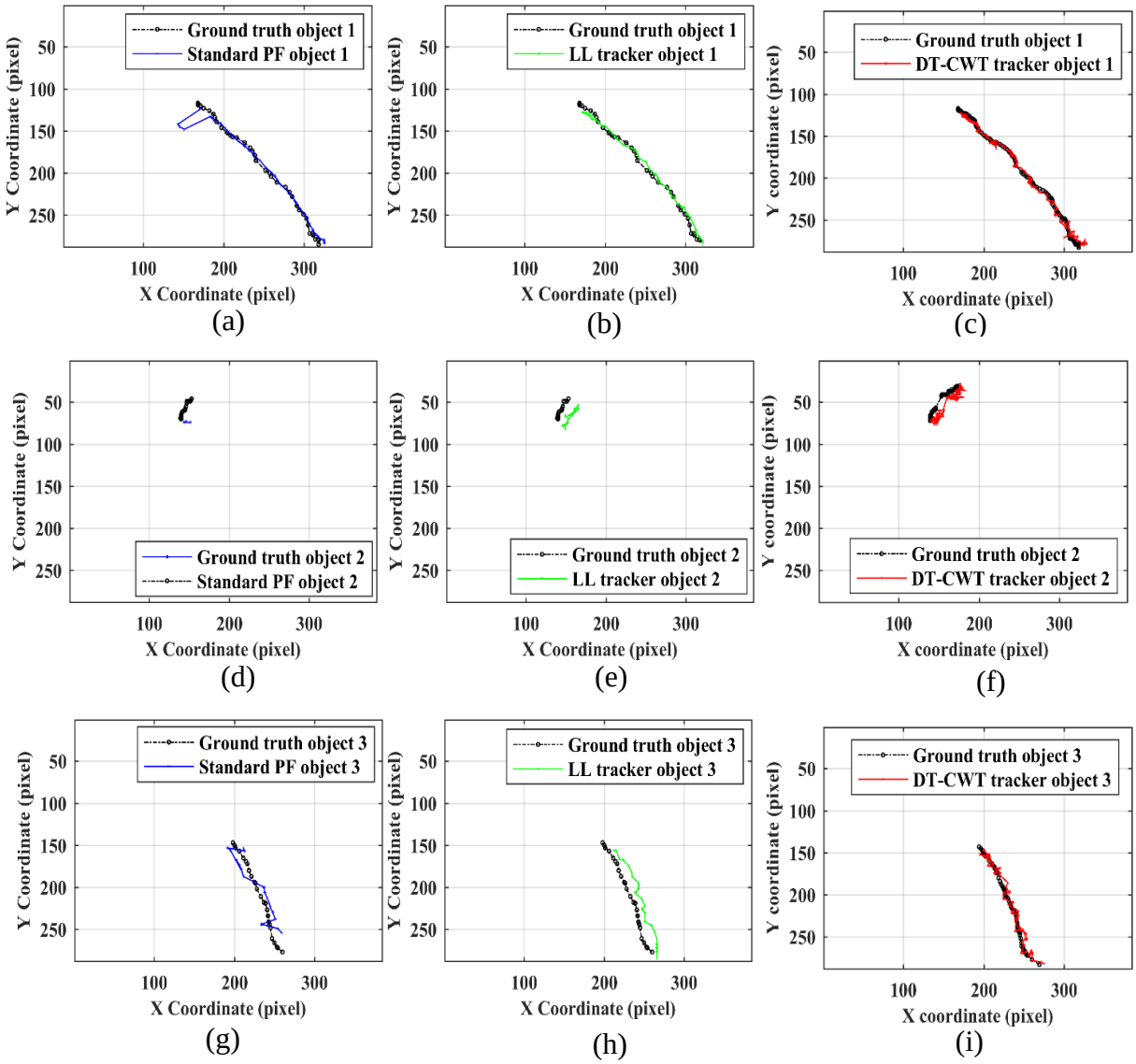
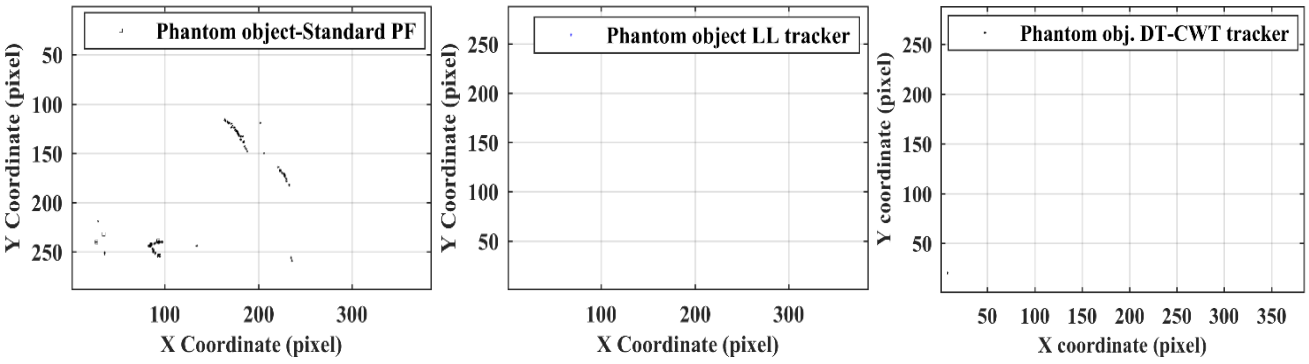


Figure 6.13. Position tracks of true objects (a) - (i) in the *"Meet_WalkTogether2"* video using a standard full resolution particle filter-based tracker (right column), a single wavelet subband $(LL)_2$ based tracker (middle column), and our multi-scale DT-CWT based tracker (left column)



(a) (b) (c)

Figure 6.14. Position tracks of phantom objects generated by: (a) the standard full resolution particle filter-based tracker, (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

Table 6.3 shows that object 1 in this video appeared in 115 *detection frames*, with cumulative track errors of 737 pixels, 765 pixels, 696 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our DT-CWT based tracker, respectively. Object 2 in this video appeared in 8 *detection frames*, with cumulative track errors of 72 pixels, 101 pixels, 63 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) DT-CWT based tracker, respectively. Object 3 in this video appeared in 58 *detection frames*, with cumulative track errors of 703 pixels, 992 pixels, 356 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our DT-CWT based tracker, respectively.

These values are a solid demonstration of the superior performance and robustness of our multi-scale tracker compared to the other two trackers.

Table 6.3. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/827 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/827 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	81	6.4	9.04	12	7.79	3.91	3.1	122
$(LL)_2$ subband tracker	62	6.6	12.7	17.05	4.43	2.45	3.74	0
DT-CWT based tracker	33	6	7.8	6.15	3.31	2.60	2.77	1

We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than those generated by the standard full-resolution particle filter-based tracker and the LL-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 81, 62, and 33 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker, respectively.

6.3.3.2 Demonstrating challenging video conditions

Presence of illumination change: Figure 6.15 (a), Figure 6.15 (b), and Figure 6.15 (c) depict the binary frames generated from the 31st video frame using the full-resolution

frame, subband $(LL)_2$, and one of the chosen subbands in our multi-scale DT-CWT based tracker, respectively.

(a) (b) (c)

Figure 6.15. Binary frames generated from the 67th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) a chosen subband by multi-scale DT-CWT based tracker

(a) (b) (c)

Figure 6.16. Visual tracking results for the 67th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

We note that the green box in Figure 6.15 (a) highlights an artifact due to the illumination change in the video frame. Figure 6.16 (a), Figure 6.16 (b), and Figure 6.16 (c) show visual tracking results, superposed on the 67^{th} video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker.We note our multi-scale tracker overcame the effect of sudden illumination change in this 67^{th} video frame.

Objects of different sizes: Figure 6.17 (a), Figure 6.17 (b), and Figure 6.17 (c) show the binary frames generated from the 200^{th} video frame using the full-resolution frame, subband $(LL)_2$, and one of the chosen subbands in our multi-scale DT-CWT based tracker, respectively. We note that the object sizes in Figure 6.9 (c) are closer to each other than the object sizes in Figure 6.17 (c) are closer to each other than the object sizes in Figure 6.17 (a). Figure 6.18 (a), Figure 6.18 (b), and Figure 6.18 (c) show visual tracking results, superposed on the 200^{th} video frame, generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale DT-CWT based tracker, respectively.

We note that, due to the presence of a large object, the standard full resolution particle filter-based tracker failed to track the smaller object. Also, the single wavelet subband $(LL)_2$ based tracker failed to track the small object due to using only one subband in a fixed scale: the second scale. Conversely, our multi-scale DT-CWT based tracker was able to overcome these problems and successfully tracked both the large and small objects.

(a) (b) (c)

Figure 6.17. Binary frames generated from the 200th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) a chosen subband by multi-scale DT-CWT based tracker

(a) (b) (c)

Figure 6.18. Visual tracking results for the 200th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale DT-CWT based tracker

6.4 Chapter summary

We developed a robust multi-scale visual tracker of multiple objects in a video using the dual-tree complex wavelet transform. A captured video frame was represented as different subbands using DT-CWT, and N independent particle filters were then applied to a small subset of these subbands. The choice of these subbands changed adaptively with each captured frame. Finally, we fused the position tracks obtained from these particle filters in order to determine the final position tracks of multiple moving objects in the video. To demonstrate the robustness of our visual tracker, we applied it to videos with

challenging visual conditions. Compared to a standard full-resolution particle filter-based tracker and a single wavelet subband $(LL)_2$ based tracker, our tracker demonstrated significantly more accurate tracking ability.

Chapter 7

Robust Tracking of Multiple Objects in Video by Adaptive Fusion of a Variable Number of Frame Wavelet Packet Subbands

7.1 Introduction

In this chapter, we develop a robust multi-scale visual tracker that adaptively fuses data from a variable number of frame wavelet packet subbands.

In our cross-section particle filter based tracker described in Chapter 5, we represented the current *difference frame* in a biorthogonal wavelet *non-redundant* dictionary, before fusing data from a fixed number, *N*, of current *difference frame* wavelet subbands. In this chapter, we generalize our design in Chapter 5 by 1) representing the current *difference frame* in a wavelet packet *redundant* dictionary; 2) fusing data from a *variable number* of wavelet packet subbands that are selected by the *Fast Best Basis Selection* algorithm [121]. This algorithm is used to obtain an accurate but *sparse* approximation for the *difference-frame* in a *redundant* dictionary with a tree structure; 3) using an explicit wavelet thresholding based denoising technique for reducing white noise in the current video frame.

The main objective of these design generalizations is to obtain a more accurate representation of the current *difference frame*, compared to the representation obtained by our earlier design, with the least number of coefficients. In other words, we aim to obtain an accurate *sparse* representation of a current *difference frame*, instead of its earlier coarse approximation obtained by keeping an arbitrarily chosen number, *N*, of its wavelet

subbands with highest energy densities. Multiple object tracking using this accurate sparse representation of a current *difference frame* should result in more robust tracking performance, particularly in the presence of challenging video conditions and events, as we would have avoided discarding potentially useful information from the current *difference frame*.

Our design generalizations mentioned above would obtain an accurate but sparse representation of a current difference frame because representing a *difference frame* in a *redundant* dictionary, i.e., wavelet packet, and using the *Fast Best Basis Selection* algorithm [121] to obtain its accurate approximation in a *non-redundant* dictionary would optimize the sparsity of such representation. In addition, using an explicit wavelet thresholding based denoising technique for reducing white noise in the current video frame would lead to a more accurate representation, compared to keeping an arbitrarily chosen number, *N*, of its wavelet subbands with highest energy densities.

After obtaining an accurate representation of a current *difference frame* that is represented by a variable number of wavelet packet subbands, a number that could change from one video frame to another, we apply a cross-section particle filter (Section 5.2.3) to all selected wavelet packet subbands. Finally, we fuse the object position tracks resulting from processing all selected wavelet packet subbands to obtain final position tracks of multiple moving objects in the video sequence.

This chapter is organized as follows: Section 7.2 describes signal denoising using wavelet thresholding techniques, Section 7.3 gives an overview of the *wavelet packet* transform and Section 7.4 describes the *Fast Best Basis Selection* algorithm. Section 7.5

provides a performance evaluation of our multi-scale wavelet packet based tracker. Finally, Section 7.6 presents a chapter summary.

7.2 Signal denoising using wavelet thresholding techniques

Signal denoising in the wavelet domain is usually simple. Typically, it could be implemented using nonlinear thresholding techniques. Wavelet coefficients with small absolute values are typically dominated by noise, while wavelet coefficients with large absolute values carry more signal information than noise. Therefore, one form of wavelet thresholding based signal denoising is to set to zero the wavelet coefficients of the signal whose absolute value is below a certain threshold.

7.2.1 Threshold selection

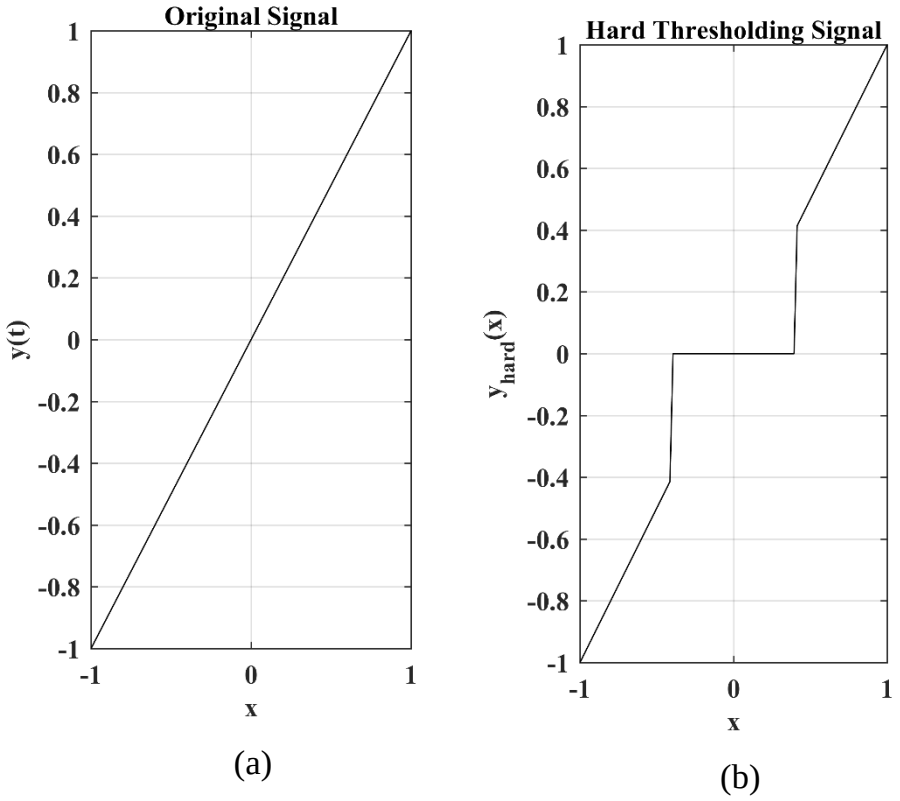
Selection of the threshold value for signal denoising is important. A small threshold value may produce a result close to the input signal, but could still be noisy. A large threshold value may produce a smooth result, but could result in a loss of details in the original signal. Also, there are different types of thresholding techniques including *hard thresholding* and *soft thresholding*. In our wavelet packet based tracker described in this chapter, we used a *hard thresholding* technique as it ensures better preservation of signal edges, compared to using *soft thresholding* [122, 123].

7.2.2 Hard thresholding

Hard thresholding for signal denoising was introduced by David L. Donoho [124]. Absolute values of the wavelet coefficients of a noisy signal are compared with a threshold value. As given by Eq. (7.1), if these wavelet coefficients of a signal, y(x), are less than the

threshold value, τ, they will be replaced by zero. Otherwise, they would be kept intact. Figure 7.1 graphically demonstrates this *hard thresholding* operation.

$$y_{hard}(t) = \begin{cases} x & |x| \geq \tau \\ 0 & |x| < \tau \end{cases} \tag{7.1}$$



(a) (b)

Figure 7.1. Hard thresholding operation, assuming the input signal was normalized to the range of [-1, 1] and the threshold level was set to τ =0.4: (a) original signal; (b) hard thresholded signal

7.3 Wavelet packet transform

The wavelet packet transform (WPT) was introduced by Coifman [125] as a generalization of the wavelet transform. The wavelet packet transform allows a richer range of possibilities for signal or image analysis. As shown in Figure 4.3, the wavelet transform

divides a signal into *approximation* and *detail* subbands using low pass and high pass filters, respectively. The resulting *approximation* subband(s) is (are) divided further into higher-level *approximation* subbands and *detail* subbands. On the other hand, in the *wavelet packet transform* both *approximation* subband(s) as well as *detail* subband(s) are divided further.

Another way to view the *wavelet packet* transform is that it generates a *redundant* dictionary, D, that is constructed by recursively splitting vector spaces $(\mathrm{W})_d^l$, at depth, d , and position, l, into q orthogonal subspaces up to some maximum recursion depth. We note that for one-dimension signal, $q = 2$, i.e., a binary tree, and two-dimension signal $q = 4$, i.e., a quad tree.

Figure 7.2 shows a quadtree representing a two-dimension *wavelet packet* transform. Each node is associated with vector subspace $(\mathrm{W})_d^l$ where d and l are the depth and position respectively. The q children of each node correspond to an orthogonal partition of W_d^l into q orthogonal subspaces, $(\mathrm{W})_{d+1}^{ql+i}$, at depth $d + 1$, and position $ql + i$, where $0 \leq i > q$ [66]. This tree-structured *redundant* dictionary is the union of all these *non-redundant* orthonormal bases of all the subspaces $(\mathrm{W})_d^l$, where a *non-redundant* orthonormal bases is constructed for each subspace $(\mathrm{W})_d^l$.

The full wavelet packet tree provides a *redundant* dictionary for the whole space, $(\mathrm{W})_0^0$, that can promote a sparse representation of any signal belonging to this space. The number of different *non-redundant* dictionaries, i.e., bases, for $(\mathrm{W})_0^0$, in this *redundant* dictionary, represented by a full *wavelet packet* quad-tree of depth, j, is equal to the number

of its *admissible subtrees*. An *admissible subtree* is a subset of the full tree, where each node is either a leaf or has q children [66].

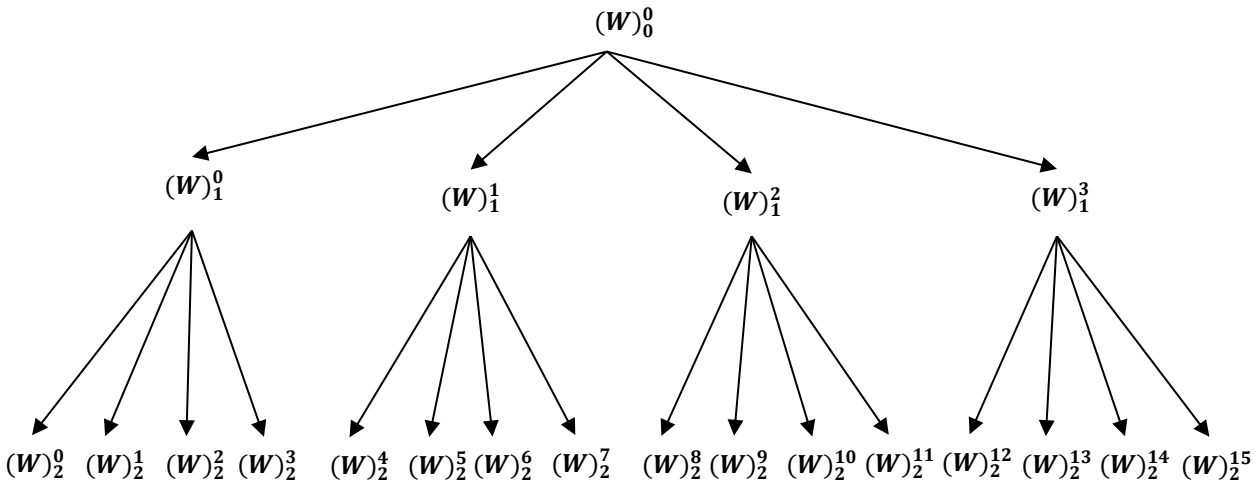


Figure 7.2. Quad-tree representing a two-dimension wavelet packet transform

7.4 *Fast Best Basis Selection* Algorithm

There is a large number of possible *non-redundant* dictionaries, i.e., bases, for $(W)_0^0$ that are given by all possible *admissible subtrees* within the full *wavelet packet* tree, which is a *redundant* dictionary. Therefore, it is possible to select within the wavelet packet tree the best *non-redundant* dictionary (basis) that will be most adapted to the time-frequency content of a given signal. Representing this signal using this best *non-redundant* dictionary (basis) would lead to its most sparse representation in the wavelet packet domain.

The *fast best basis selection* algorithm, introduced by Coifman and Wickerhauser [65, 126], selects the best *non-redundant* dictionary (basis) from a *redundant* tree-structured dictionary, e.g., a wavelet packet tree, that will be most adapted to the time-frequency content of a given signal. This fast selection algorithm relies on the tree structure of the

redundant dictionary, and it involves an optimization problem that minimizes an additive information cost function, e.g., Shannon entropy [127].

Table 7.1 describes the algorithm for *Fast Best Basis Selection* from *redundant* tree dictionaries. We note that this algorithm is practically applicable, i.e., numerically stable, for tree dictionaries whose nodes $(\mathrm{W})_d^l$ represent subspaces with orthonormal bases. Therefore, in our visual tracker described in this chapter, we used the *Haar* orthogonal wavelets to generate our wavelet packet redundant tree dictionary.

Table 7.1. *Fast Best Basis Selection* from *redundant* tree dictionaries

Initialization:

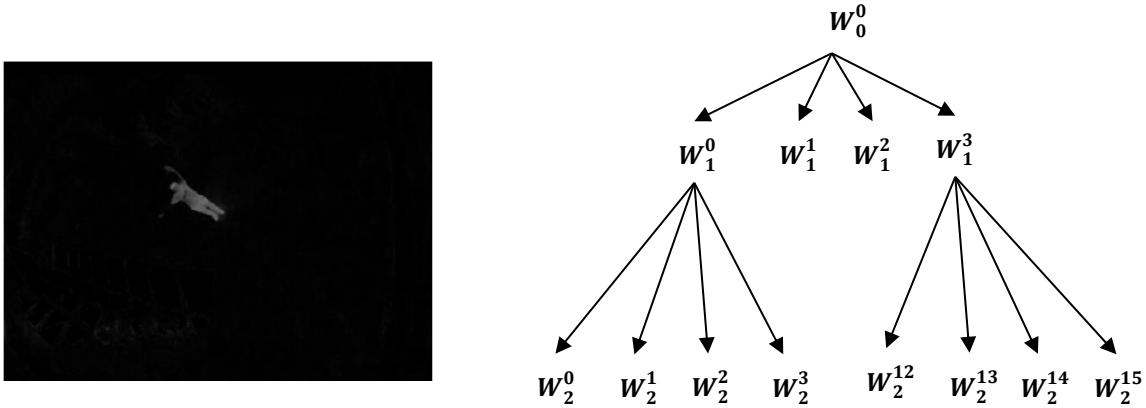
- Construct the full wavelet packet tree, T, and let $W_{l,k}$ be the set of subspace basis vectors, $0 \leq l < L$, $1 \leq k \leq 2^l - 1$.
- Set the initial basis selection to all the leaf nodes:

$$W = \{W_{L-1,1}, W_{L-1,2}, \dots, W_{L-1,k}, \dots, W_{L-1,2^{L-1}}\}$$

Repeat from the leaf level of the tree to its root:

- Compute the entropy of a parent node $(E_{l,k})$ and its children $(E_{l+1,2k}, E_{l+1,2k-1})$.
- Compare the entropy of the parent node, $E_{l,k}$, with the sum of the entropies of its two children nodes $(E_{l+1,2k} + E_{l+1,2k-1})$, if the entropy of the parent is greater than the sum of entropies of its children, replace $W_{l+1,2k}$ and $W_{l+1,2k-1}$ with $W_{l,k}$ in W.

Figure 7.4. shows an example of a result obtained by the *Fast Best Basis Selection* algorithm for representing the 10th *difference frame* in the *''OneLeaveShopReenter2front"* video sequence.



(a) (b)

Figure 7.3. Example of a result obtained from the fast best basis selection algorithm (a) 10th *difference frame* in the *''OneLeaveShopReenter2front"* video sequence represented by its best wavelet packet basis; (b) best wavelet packet basis for this 10th difference frame.

7.5 Performance evaluation of our multi-scale WPT based tracker

In the following examples, we show that our multi-scale WPT visual tracker overcame the presence of challenging conditions in four video sequences and demonstrated better tracking performance compared to a standard full resolution particle filter-based tracker and compared to a single wavelet subband $(LL)_2$ based tracker, results obtained from our multiscale tracker demonstrate significantly better tracking performance.

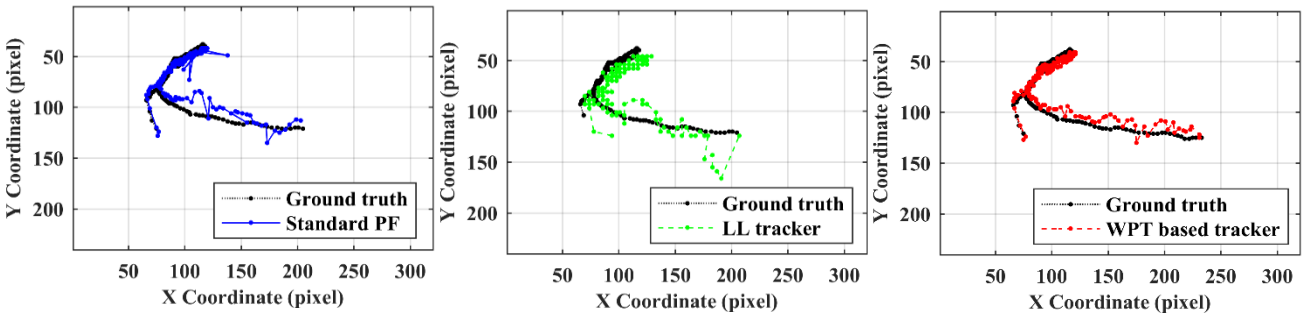
7.5.1 Example demonstrating object shadow and partial object camouflage

To demonstrate the improved performance of our multi-scale WPT based tracker, compared to a standard full resolution particle filter-based tracker and compared to a single

wavelet subband $(LL)_2$ based tracker, we used *"Intelligentroom_raw"* video sequence that included the presence of object shadow and partial object camouflage.

7.5.1.1 Comparison of resulting position tracks

Figure 7.5 shows the position tracks of objects obtained using standard particle filter-based visual tracker, single wavelet subband $(LL)_2$ based tracker and our multi-scale WPT based tracker. Figure 7.5 (a), Figure 7.5 (b), and Figure 7.5 (c) show the true position tracks of the object, in addition to ones generated by the standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale WPT based tracker, respectively.

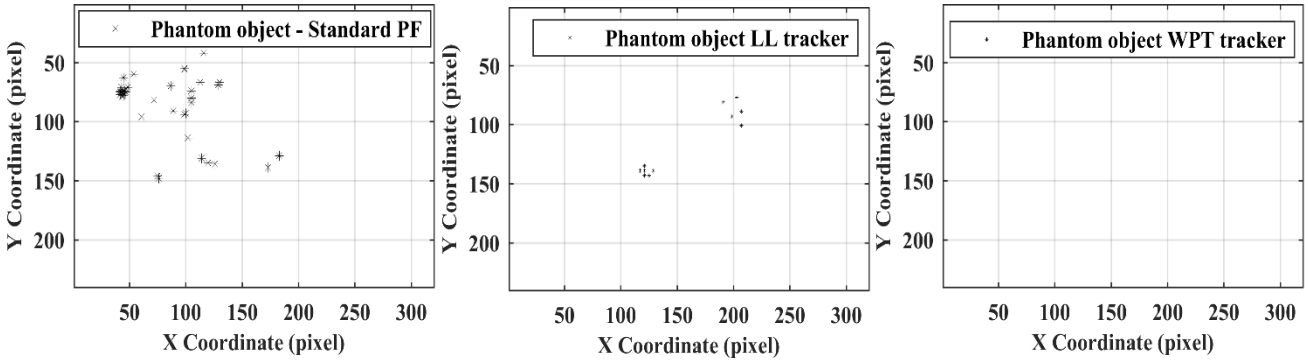


(a) (b) (c)

Figure 7.5. Position tracks of true objects in the video *"Intelligentroom_raw"* using: (a) a standard full-resolution particle filter-based tracker; (b) a single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker.

Figure 7.6 shows position tracks of phantom objects generated by the standard full resolution particle filter-based tracker and single wavelet subband $(LL)_2$ based tracker. These phantom objects could appear due to the presence of object's shadow, and partial object camouflage. We note that our multiscale WPT based tracker generated no phantom

objects, while the standard full resolution particle filter-based tracker and single wavelet subband $(LL)_2$ based tracker generated many. This is a further demonstration of the robustness of our multi-scale tracker.



(a) (b) (c)

Figure 7.6. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker.

As shown in Table 7.2, the object in this video appeared in 214 *detection frames*, with cumulative track errors of 987.2 pixels, 2032 pixels, 1009.1 pixels using 1) standard full resolution particle filter, 2) $(LL)_2$ subband based tracker, and 3) our multi-scale WPT based tracker, respectively.

We note that due to the presence of challenging conditions and unexpected events in this video sequences, e.g., object shadow; partial object camouflage; and low signal-to-noise ratio, standard full resolution particle filter-based tracker generated 90 phantom objects, while our WPT multi-scale based tracker overcame the presence of these challenging conditions and unexpected events and generated no phantom objects.

Table 7.2. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/300 frames)	Average position track error (pixel/*detection frame*)	Standard deviation of tracking errors	Phantom object (event/300 frames)
Full resolution PF tracker	0	4.635	4.51	90
$(LL)_2$ subband tracker	2	9.539	5.85	11
WPT based tracker	0	4.7	2.61	0

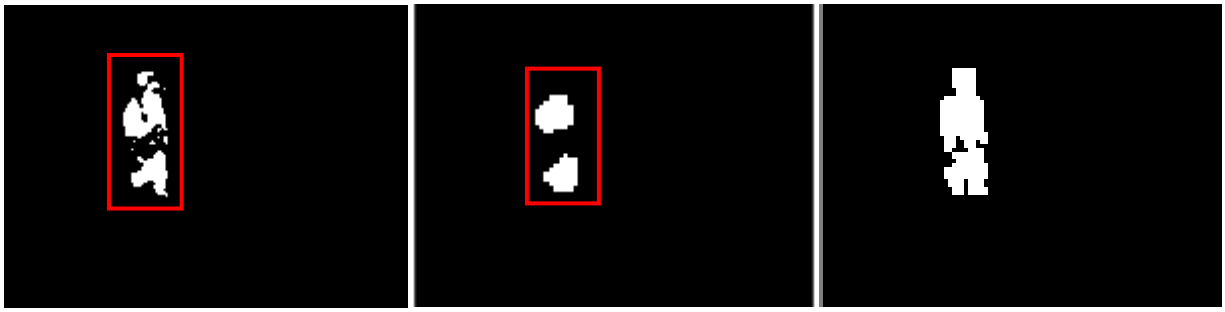
These values are a solid demonstration of the superior performance and robustness of our WPT multi-scale based tracker compared to the other two trackers.

7.5.1.2 Demonstrating challenging video conditions

Partial object camouflage: Figure 7.7 (a), Figure 7.7 (b), and Figure 7.7 (c), show the binary frames generated from the 265^{th} video frame using the full-resolution frame, subband $(LL)_2$, and one subband frame from the constructed best wavelet packet tree for this particular frame in our multi-scale tracker, respectively. We note that the red boxes in Figure 7.7 (a), Figure 7.7 (b) highlight the division of an object into two due to the presence of partial object camouflage.

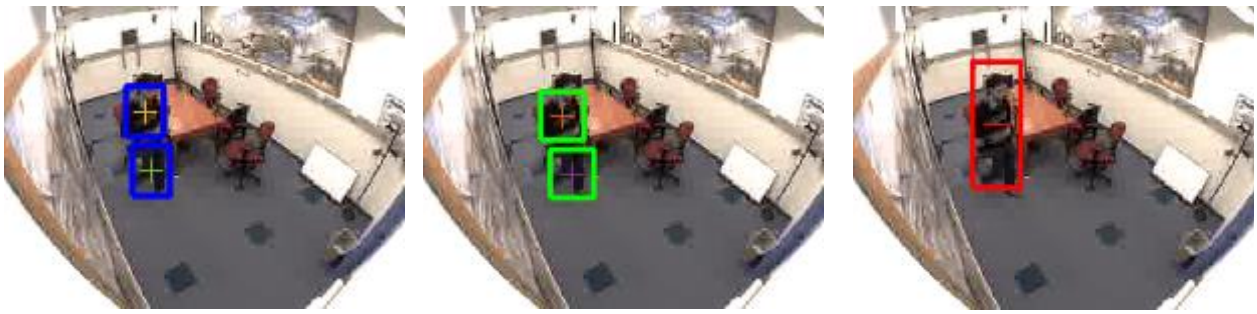
Figure 7.8 (a), Figure 7.8 (b), and Figure 7.8 (c) show visual tracking results, superposed on the 265^{th} video frame, generated by a standard full resolution particle filter-

based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that standard full resolution particle filter-based based tracker and single wavelet subband $(LL)_2$ based tracker generated two phantom objects due to the object division in Figure 7.7 (a), Figure 7.7 (b), while our multi-scale tracker overcame the presence of partial object camouflage in this 265th video frame.



(a) (b) (c)

Figure 7.7. Binary frames generated from the 265th frame using: (a) the full-resolution frame; (b) subband $(LL)_2$; (c) a subband frame from the constructed best wavelet packet tree



(a) (b) (c)

Figure 7.8. Visual tracking results for 265th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

Object shadow: Figure 7.9 (a), Figure 7.9 (b), and Figure 7.9 (c) show the binary frames generated from the 240th video frame using the full resolution frame, using subband

$(LL)_2$ subband, and using one subband frame from the constructed best wavelet packet tree, respectively. We note that the green box in Figure 7.9 (a) highlights an artifact due to the presence of object shadow.

(a) (b) (c)

Figure 7.9. Binary frames generated from the 245th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree

(a) (b) (c)

Figure 7.10. Visual tracking results for 245th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

Figure 7.10 (a), Figure 7.10 (b), and Figure 7.10 (c), show visual tracking results, superposed on the 245th video frame, , generated by a standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale WPT based

tracker, respectively. We note that standard full resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 7.10 (a) while single wavelet subband $(LL)_2$ based tracker and our multi-scale WPT based tracker overcame the presence of object shadow in this 245th video frame.

7.5.2 Example demonstrating background motion, object shadow, and partial object camouflage

To demonstrate the improved performance of our multi-scale WPT based tracker, compared a standard full resolution particle filter-based tracker and compared to a single wavelet subband $(LL)_2$ based tracker, we applied it to *"OneLeaveShopReenter2front"* video sequence that included object shadow and partial object camouflage.

7.5.2.1 Comparison of resulting position tracks

Figure 7.11 shows the position tracks of objects obtained using standard particle filter-based visual tracker, single wavelet subband $(LL)_2$ based tracker and our multi-scale tracker. Figure 7.11 (a) - (c), Figure 7.11 (d) - (f), and Figure 4.10 (g) - (j) show the true position tracks of the three objects, in addition to ones generated by the standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that the differences between the position paths generated by our multi-scale WPT based tracker and the true position paths are significantly smaller than when a standard full resolution particle filter-based tracker, and single wavelet subband $(LL)_2$ based tracker were used.

Figure 7.12 shows position tracks of phantom objects generated by the standard full resolution particle filter-based tracker. These phantom objects could appear due to the presence of object shadows, and partial object camouflage. We note that while the standard full resolution particle filter-based tracker generated many phantom objects, single wavelet subband $(LL)_2$ based tracker generated two. Our multi-scale WPT based tracker could use more subbands than our previous trackers. Therefore, it detects object boundaries in different directions which yield to detection of the background motion as a phantom object.

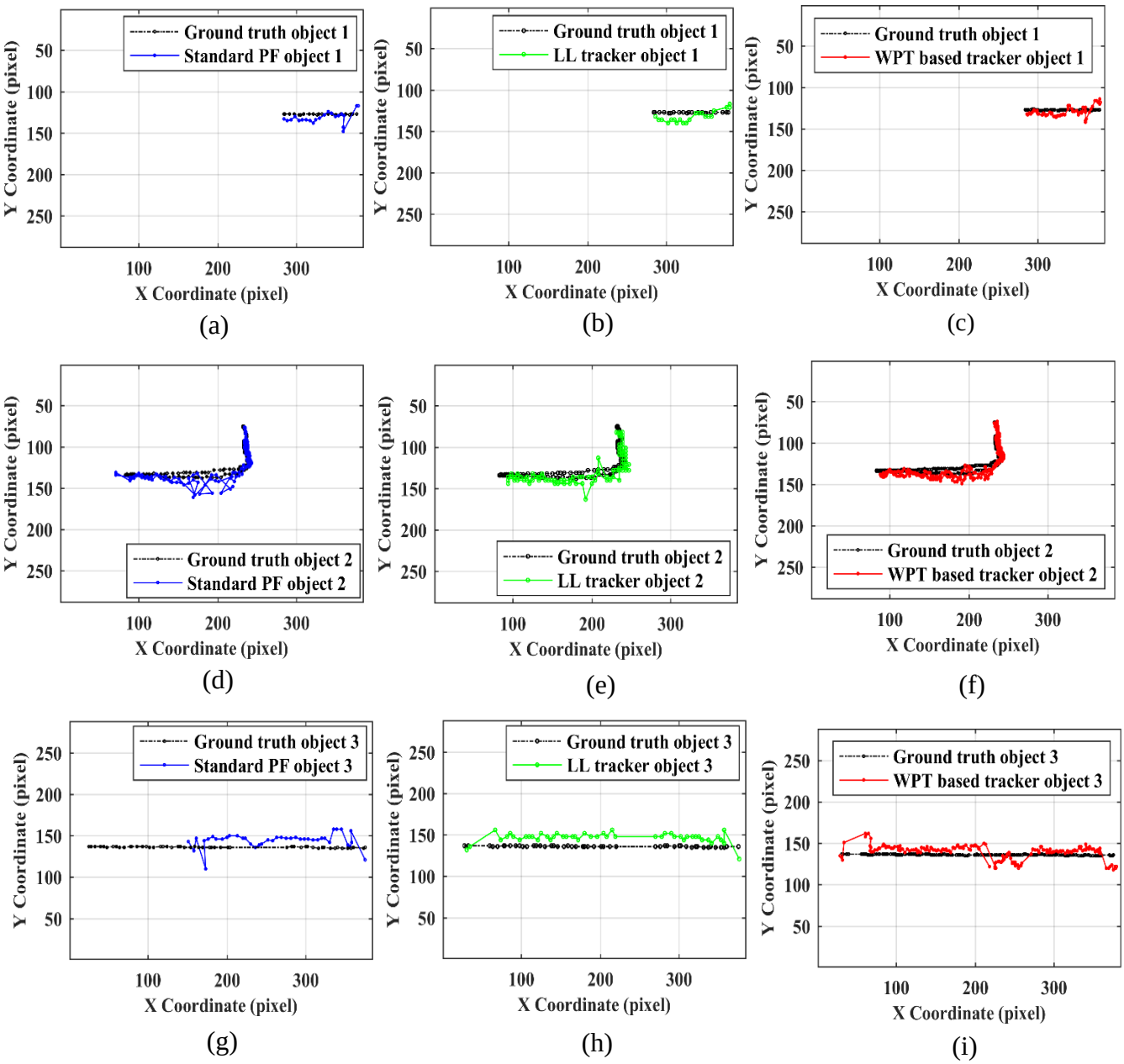


Figure 7.11. Position tracks of true objects (a) - (i) in the *''OneLeaveShopReenter2front''* video using a standard full-resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multi-scale WPT based tracker (left column)

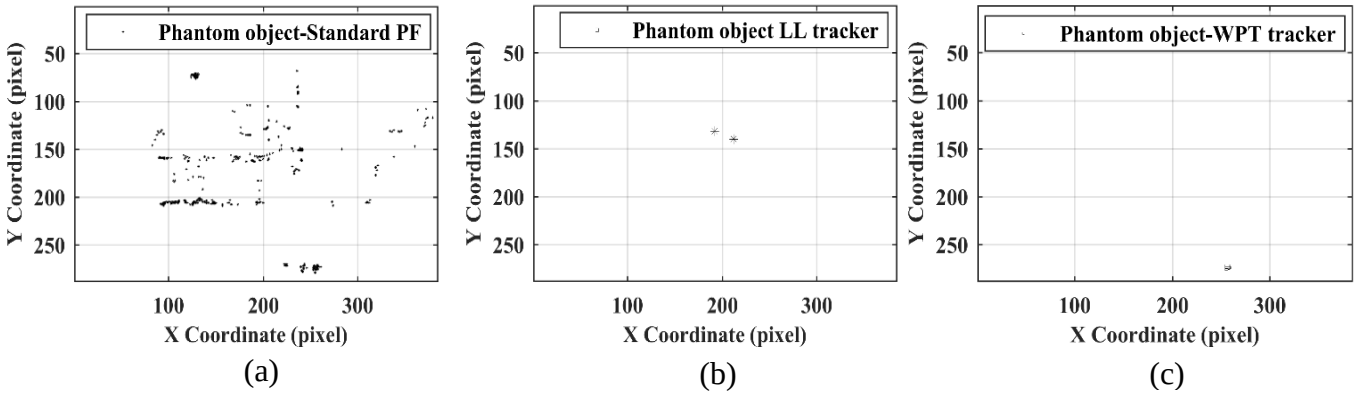


Figure 7.12. Position tracks of phantom objects generated by: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker; and (c) our multi-scale WPT based tracker

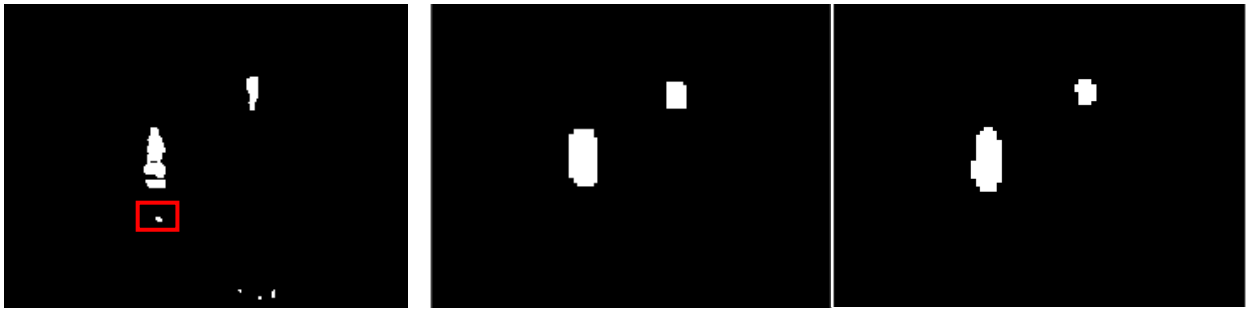
As shown in Table 7.3, object 1 in this video appeared in 58 detection frames, with cumulative track errors of 345 pixels, 468 pixels, 327 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale WPT based tracker, respectively. Object 2 in this video appeared in 477 *detection frames*, with cumulative track errors of 1918 pixels, 3450 pixels, 2107 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale WPT based tracker, respectively. Object 3 in this video appeared in 122 *detection frames*, with cumulative track errors of 1610 pixels, 1530 pixels, 906 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multi-scale WPT based tracker, respectively. These values are a solid demonstration of the superior performance and robustness of our multiscale tracker compared to the other two trackers.

Table 7.3. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/558 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/558 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	55	6.56	3.99	13.18	3.63	3.12	5.58	469
$(LL)_2$ subband tracker	80	8.17	7.24	12.47	3.84	2.97	3.4	2
WPT based tracker	23	5.32	4.57	6.65	2.68	2.37	2.82	0

7.5.2.2 Demonstrating challenging video conditions

Object shadow: Figure 7.13 (a), Figure 7.13 (b) and Figure 7.13 (c) show the binary frames generated from the 116^{th} video frame using the full resolution frame, using subband $(LL)_2$ and using one subband frame from the constructed best wavelet packet tree, respectively. We note that the red box in Figure 7.13 (a) highlights an artifact due to the presence of object shadow.



(a) (b) (c)

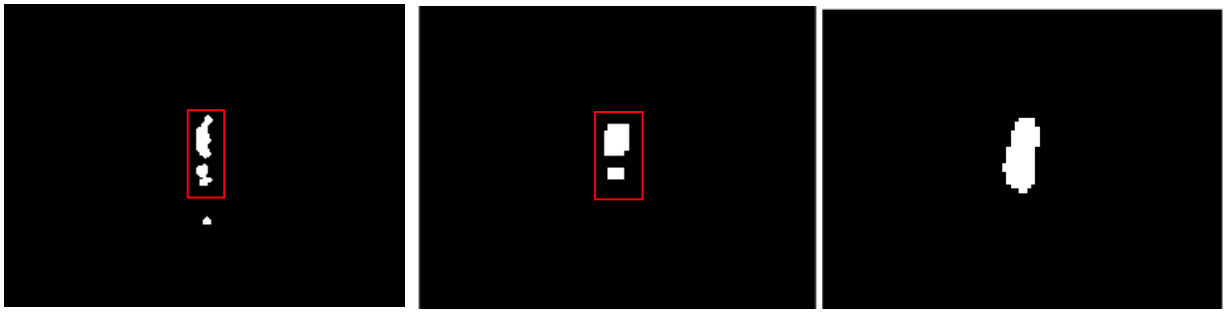
Figure 7.13. Binary frames generated from the 116th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree

(a) (b) (c)

Figure 7.14. Visual tracking results for the 116th video frame using; (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

Figure 7.14 (a), Figure 7.14 (b), and Figure 7.14 (c) show visual tracking results, superposed on the 116th video frame, generated by a standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that standard full resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 7.13 (a), while our multi-scale tracker overcame the presence of object shadow in this 116th video frame.

Partial object camouflage: Figure 7.15 (a), Figure 7.15 (b), and Figure 7.15 (c) show the binary frames generated from the 427th video frame using the full resolution frame, using subband $(LL)_2$, and using one subband frame from the constructed best wavelet packet tree, respectively. We note that the red boxes in Figure 7.15 (a), and Figure 7.15 (b) highlight the division of an object into two due to the presence of partial object camouflage.



(a) (b) (c)

Figure 7.15. Binary frames generated from the 427th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree

Figure 7.16 (a), Figure 7.16 (b), and Figure 7.16 (c) show visual tracking results, superposed on the 427th video frame, generated by a standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale WPT based tracker, respectively. We note that standard full resolution particle filter-based tracker generated two phantom objects due to the presence of the object division in Figure 7.16 (a) and object shadow, also single wavelet subband $(LL)_2$ based tracker generated a phantom object due to the presence of the object division in Figure 7.16 (b) while our multi-scale tracker overcame the presence of partial object camouflage in this 427th video frame.

(a) (b) (c)

Figure 7.16. Visual tracking results for the 427th video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

7.5.3 Example demonstrating the presence of objects with different sizes and sudden illumination change

To demonstrate the improved performance of our multi-scale WPT based tracker, compared a standard full resolution particle filter-based tracker and compared to a single wavelet subband $(LL)_2$ based tracker, we applied it to the *"Meet_WalkTogether2"* video sequence that included object shadow and partial object camouflage.

7.5.3.1 Comparison of resulting position tracks

Figure 7.17 shows the position tracks of objects obtained using standard particle filter-based visual tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale subband particle filters tracker. Figure 7.17 (a) - (c), Figure 7.17 (d) - (f), and Figure 7.17 (g) -- (j) show the true position tracks of the three objects, in addition to ones generated by

the standard full-resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker and our multi-scale WPT based tracker, respectively.

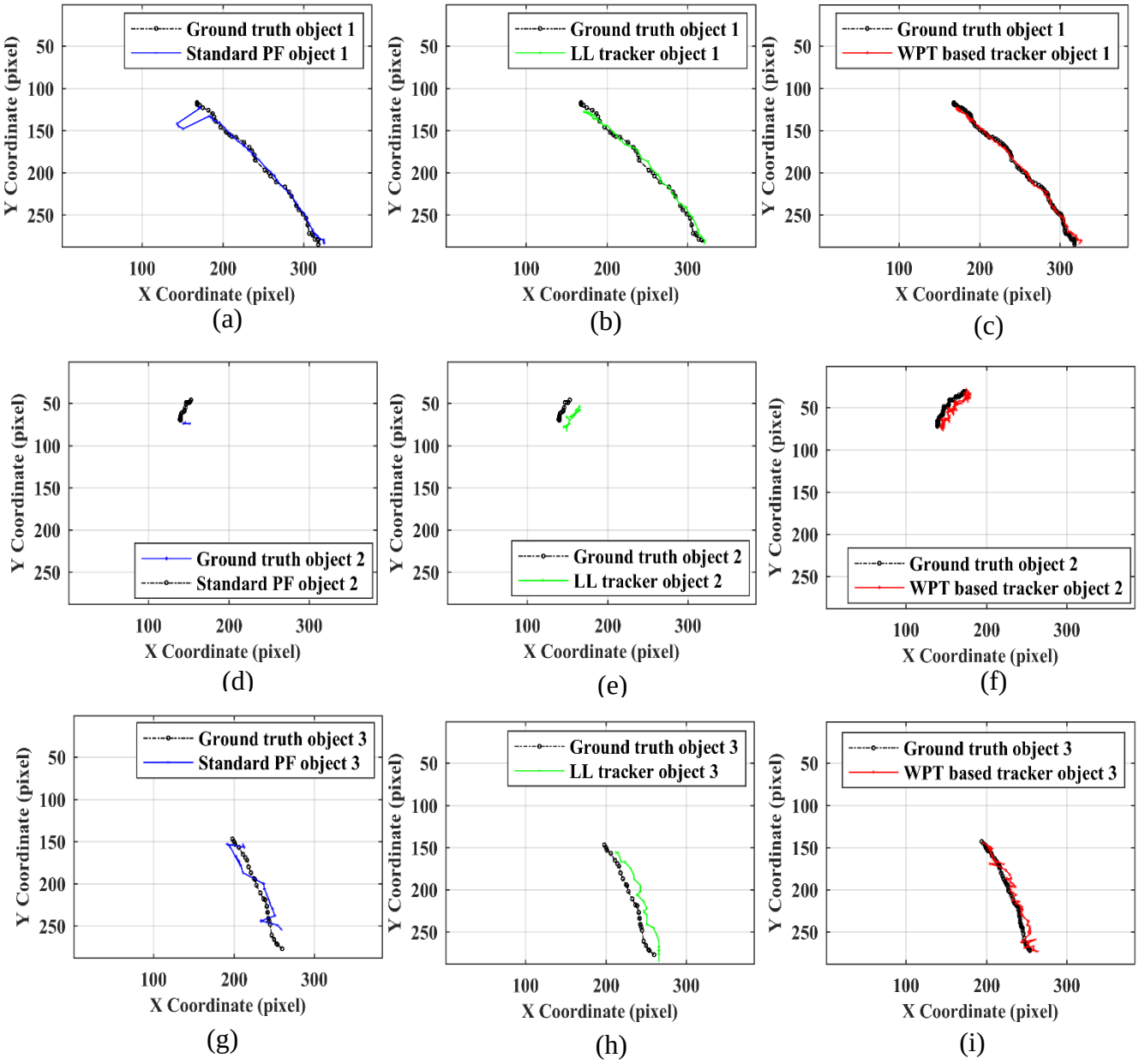
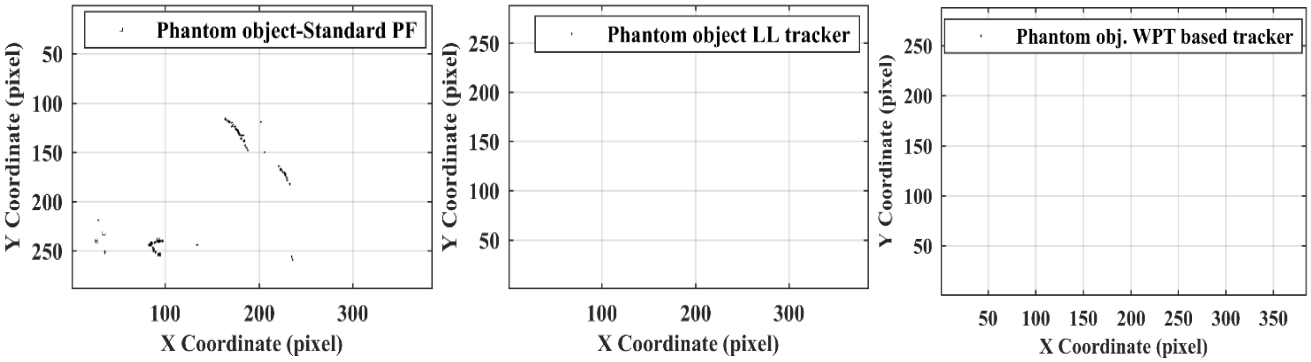


Figure 7.17. Position tracks of true objects (a) - (i) in the *"Meet_WalkTogether2"* video using the standard full-resolution particle filter-based tracker (right column), the single wavelet subband $(LL)_2$ based tracker (middle column), and our multi-scale WPT based tracker (left column)



(a) (b) (c)

Figure 7.18. Position tracks of phantom objects generated by: (a) the standard full resolution particle filter-based tracker, (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than when a standard full resolution particle filter-based tracker and single wavelet subband $(LL)_2$ based tracker were used.

Figure 7.18 show position tracks of phantom objects generated by the standard full resolution particle filter-based tracker. These phantom objects could appear due to the presence of background motion, object shadows, and partial object camouflage. We note the standard full resolution particle filter-based tracker generated many. Our multi-scale WPT based tracker could use more subbands than our previous trackers. Therefore, it detects object boundaries in different directions which yield to detection of the background motion as a phantom object.

As shown in Table 7.4, object 1 in this video appeared in 114 *detection frames*, with cumulative track errors of 737 pixels, 755 pixels, 512 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale tracker, respectively. Object 2 in this video appeared in 8 *detection frames*, with cumulative track

errors of 72 pixels, 101 pixels, 52 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale tracker, respectively. Object 3 in this video appeared in 60 *detection frames*, with cumulative track errors of 718 pixels, 1023 pixels, 450 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale tracker, respectively.

These values are a solid demonstration of the superior performance and robustness of our multiscale tracker compared to the other two trackers.

Table 7.4. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/827 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/827 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	81	6.4	9.04	12	7.80	3.91	3.199	122
$(LL)_2$ subband tracker	62	6.6	12.7	17.05	4.47	2.45	3.70	0
WPT based tracker	33	4.4	6.6	7	2.88	1.31	3.46	15

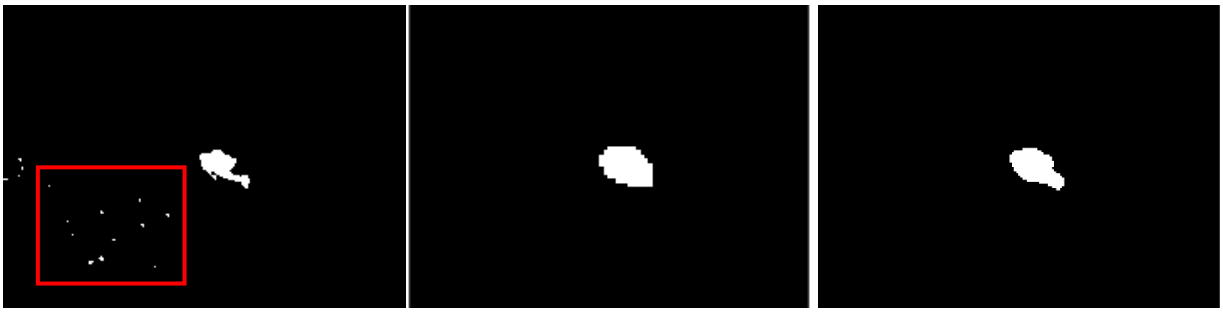
We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than those generated by the standard full-resolution particle filter-based tracker and the $(LL)_2$-based tracker. Moreover, we note that

the number of times that a real object failed to be tracked was 81, 62, and 45 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively.

7.5.3.2 Demonstrating challenging video conditions.

Presence of sudden illumination change: Figure 7.19 (a), Figure 7.19 (b), and Figure 7.19 (c) show the binary frames generated from the 45^{th} video frame using the full resolution frame, using subband $(LL)_1$ and using one subband frame from the constructed best wavelet packet tree, respectively.

We note that the red box in Figure 7.19 (a) highlights an artifact due to a sudden illumination change in this video frame. Figure 7.20 (a), Figure 7.20 (b), and Figure 7.20 (c) show visual tracking results, superposed on the 45^{th} video frame, generated by a standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively. We note that standard full resolution particle filter-based tracker generated a phantom object due to the presence of the artifact in Figure 7.19 (a), while our multi-scale tracker overcame the effect of sudden illumination change in this 45^{th} video frame.



(a) (b) (c)

Figure 7.19. Binary frames generated from the 45th frame using: (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree

(a) (b) (c)

Figure 7.20. Visual tracking results for 45th video frame using: (a) the standard particle filter full resolution based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

Objects of different sizes: Figure 7.21 (a), Figure 7.21 (b), and Figure 7.21 (c) show the binary frames generated from the 201st video frame using the full resolution frame, using subband $(LL)_2$ and using one subband frame from the constructed best wavelet packet tree, respectively. We note that the object sizes in Figure 7.21 (c) are closer to each other than the object sizes in Figure 7.21 (a). Figure 7.22 (a), Figure 7.22 (b), and Figure 7.22 (c) show visual tracking results, superposed on the 201st video frame, generated by a

standard full resolution particle filter-based tracker, single wavelet subband $(LL)_2$ based tracker and our multi-scale tracker, respectively.

(a) (b) (c)

Figure 7.21. Binary frames generated from the 201st frame using; (a) the full resolution frame; (b) subband $(LL)_2$; (c) subband frame from the constructed best wavelet packet tree

(a) (b) (c)

Figure 7.22. Visual tracking results for the 201st video frame using: (a) the standard full-resolution particle filter-based tracker; (b) the single wavelet subband $(LL)_2$ based tracker, and (c) our multi-scale WPT based tracker

We note that, due to the presence of a large object, the standard full resolution particle filter-based tracker failed to track the smaller object moreover it showed partial camouflage problem. Also, single wavelet subband $(LL)_2$ based tracker failed to track small object due to using only one subband in a fixed scale, the second scale. Our multi-scale tracker,

however, overcame these problems in this 201st video frame and was able to track both large and small objects successfully.

7.5.4 Example demonstrating presence of objects with different sizes and partial object camouflage

The The video sequence in this example, "ATCS" is from the Visor database (288 X 384 pixels, 30 fps, 1313 frames). This video sequence shows three moving people.

7.5.4.1 Comparison of resulting position tracks

Figure 7.24 shows the position tracks of objects obtained using the standard particle filter-based visual tracker, the single wavelet subband (LL)2 based tracker, and our multi-scale WPT based tracker. Figure 7.24 (a) - (c), Figure 7.24 (d) - (f), and Figure 7.24 (g) - (i) show the true position tracks of the three objects, as well as those generated by the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale WPT based tracker, respectively. We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are significantly smaller than the differences generated by the standard full-resolution particle filter-based tracker and single wavelet subband $(LL)_2$ tracker.

Figure 7.23 shows the visual tracking results for a sample of four video frames using our multi-scale tracker. We note that our multi-scale WPT based tracker generated no phantom objects, while the standard full-resolution particle filter-based tracker generated many. This is a further demonstration of our multi-scale tracker's robustness.

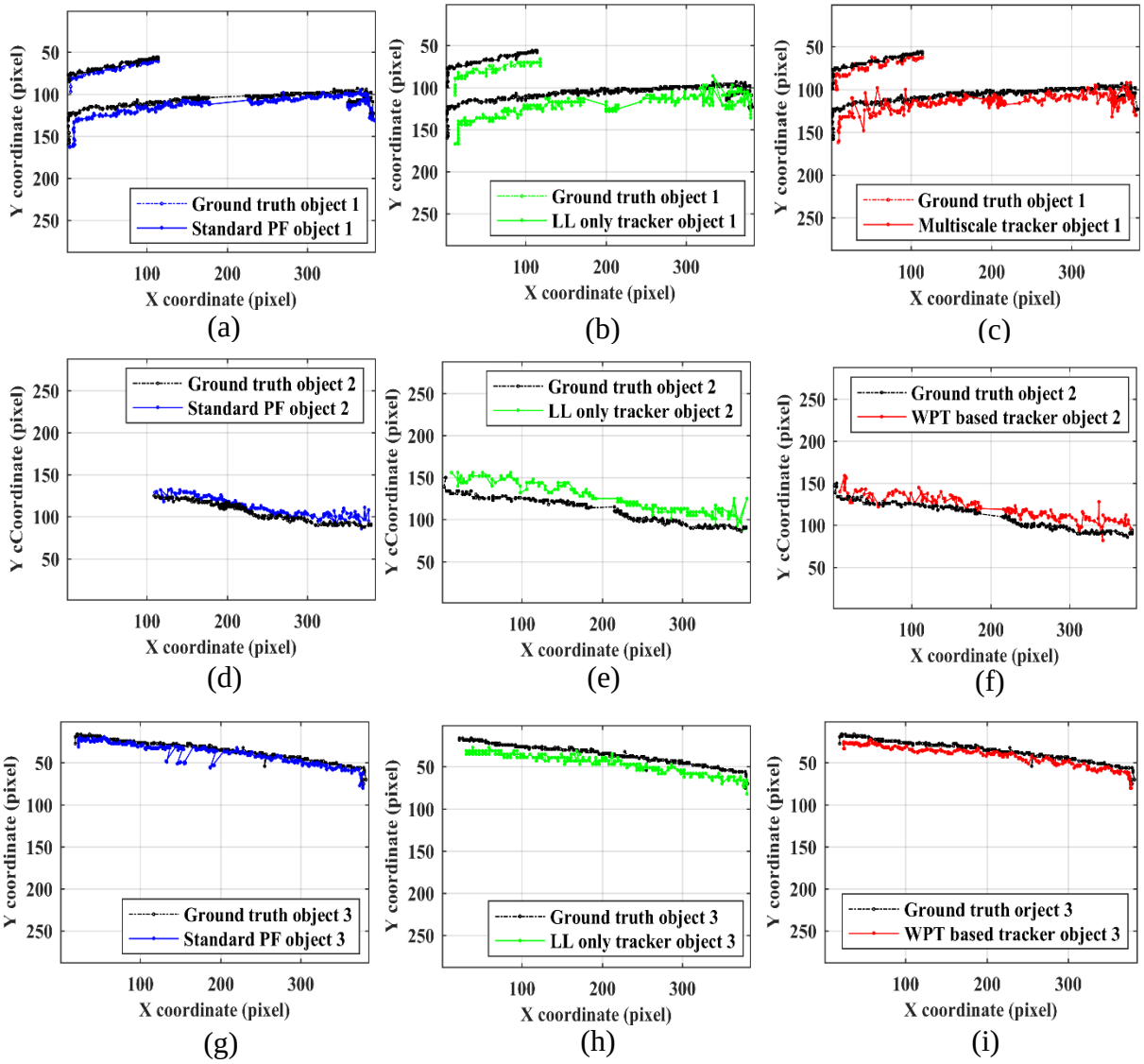


Figure 7.24. Position tracks of true objects (a) - (i) in the "*ATCS*" video using a standard full resolution particle filter-based tracker (right column), single wavelet subband $(LL)_2$ based tracker (middle column), and our multiscale WPT based tracker (left column)

Figure 7.23. Visual tracking results for four video frames using our multi-scale tracker

As shown Table 7.5, object 1 video appeared in 399 *detection frames*, with cumulative track errors of 2284 pixels, 5815 pixels, 3861 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale WPT based tracker, respectively. Object 2 in this video appeared in 157 *detection frames*, with cumulative track errors of 1159 pixels, 2472 pixels, 1935 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale WPT based tracker, respectively. Object 3 in this video appeared in 289 *detection frames*, with cumulative track errors of 1621 pixels, 3730 pixels, 2263 pixels using 1) standard full resolution particle filter, 2) single wavelet subband $(LL)_2$, and 3) our multiscale WPT based tracker, respectively. These values are a solid demonstration of the superior performance and robustness of our multiscale tracker compared to the other two trackers.

Table 7.5. Number of missed object events, average position track errors, and number of phantom object events

Visual tracker type	Missed object (event/1313 frames)	Average position track error (pixel/detection frame)			Standard deviation of track errors			Phantom object (event/1313 frames)
		Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	Obj.1	
Full resolution PF tracker	83	5.72	7.38	5.61	4.98	5.42	4.98	31
$(LL)_2$ subband tracker	34	14.57	15.76	12.90	4.93	5.42	4.87	0
WPT based tracker	3	9.67	12.32	7.83	3.74	5.69	3.57	0

We note that the differences between the position paths generated by our multi-scale tracker and the true position paths are smaller than those generated by the standard full-resolution particle filter-based tracker and the LL-based tracker. Moreover, we note that the number of times that a real object failed to be tracked was 83, 34, and 15 for the standard full-resolution particle filter-based tracker, the single wavelet subband $(LL)_2$ based tracker, and our multi-scale tracker, respectively.

7.6 Chapter summary

In this chapter, we developed a robust multi-scale visual tracker that adaptively fused data from a variable number of frame wavelet packet subbands. We generalized our design

from Chapter 5 by 1) representing the current *difference frame* in a wavelet packet *redundant* dictionary; 2) fusing data from a *variable number* of wavelet packet subbands that are selected by the *Fast Best Basis Selection* algorithm; 3) using an explicit wavelet thresholding based denoising technique for reducing white noise in the current video frame. The main objective of these design generalizations was to obtain an accurate *sparse* representation of a current *difference frame*, instead of its earlier coarse approximation obtained by keeping an arbitrarily chosen number, *N*, of its wavelet subbands with highest energy densities. Then we applied a cross-section particle filter to all the selected wavelet packet subbands. Finally, we fused the outputs of processing these wavelet packet subbands to obtain final position tracks of multiple moving objects in the video sequence. Compared to the results obtained by a standard full resolution particle filter-based tracker, and a single wavelet subband $(LL)_2$ based tracker, the results obtained from our multiscale WPT based tracker demonstrate significantly more accurate tracking performance.

Chapter 8

Conclusions and Future Work

Tracking of multiple objects in video sequences is an important research problem because of its many different applications. Significant progress has been made on this topic in the last few decades. However, the ability to track objects accurately in video sequences that have challenging conditions and unexpected events, e.g., background motion and object shadows, objects with different sizes and contrasts, a sudden change in illumination, partial object camouflage, and low signal-to-noise ratio remains a challenging research problem. In this thesis, we designed and implemented four novel robust visual trackers to overcome these challenging conditions and unexpected events.

8.1 Thesis Contributions

We developed four robust multi-scale visual trackers in the wavelet domain using Bayesian approach:

1. A robust multi-scale visual tracker that represents a captured video frame as different subbands in the wavelet domain. It then applies N independent particle filters to a small subset of these subbands, where the choice of this subset of wavelet subbands changes with each captured frame. Finally, it fuses the output tracks of these N independent particle filters to obtain final position tracks of multiple moving objects in the video sequence (tracker-fusion). Compared to a standard full-resolution particle filter-based tracker and a single wavelet subband

(LL)2 based tracker, our multi-scale tracker demonstrated significantly more accurate tracking performance, as well as reduced average frame processing times.

2. To reduce the computational cost of our first visual tracker, we also developed a robust multi-scale visual tracker that adaptively fused N frame subbands using a single cross-section particle filter. Compared to the results obtained by a standard particle filter-based tracker, our results demonstrated significantly more accurate tracking performance. Furthermore, our cross-section particle filter-based tracker required a computational cost of approximately 50% of that required by our first multi-scale tracker.

3. A robust multi-scale visual tracker for multiple objects in video using subband frames that were adaptively selected from a Dual-Tree Complex Wavelet Transform (Dt-CWT). We used the DT-CWT to avoid shortcomings of real-valued wavelet transforms, e.g., shift variance and low directional selectivity.

4. A robust multi-scale visual tracker of multiple objects in a video that used a sparse representation of a current frame in the wavelet packet domain, obtained by the Fast Best Basis Selection algorithm. Compared to a standard full-resolution particle filter-based tracker and a single wavelet subband (LL)2 based tracker, our multi-scale tracker demonstrated significantly more accurate tracking performance.

8.2 Publications

1- Ahmed Mahmoud and Sherif S. Sherif, "Robust Tracking of Multiple Objects in Video by Adaptive Fusion of Subband Particle Filters," *IET computer Vision*, July 31th, 2018.

2- Ahmed Mahmoud and Sherif S. Sherif, "Dual-tree complex wavelet transform for robust visual tracking of multiple objects in video," *11th International Conference on Electrical Engineering (ICEENG)*, Military Technical College, Cairo, Egypt, 3-5 April, 2018.

3- Ahmed Mahmoud and Sherif S. Sherif, "Robust Tracking of Multiple Objects in Video by Adaptive Fusion of *N* frame Subbands Using a Cross-section Particle Filter," to be submitted to *Computer Vision and Image Understanding*, August 2018.

4- Ahmed Mahmoud and Sherif S. Sherif, "Robust Tracking of Multiple Objects in Video by Adaptive Fusion of a Variable Number of Frame Wavelet Packet Subbands," to be submitted to *Pattern Recognition*, September, 2018.

8.3 Possible Future Work

The work described in this thesis could be extended in the following possible ways:

1. Use multiple motion models to represent maneuvering objects, whose dynamic behavior changes with time.
2. Add more visual features or cues, e.g., object contour shape, to the used particle

filter's likelihood function to increase tracking robustness.

3. Use a parallel processing platform to implement different subband particle filters based visual trackers. By allocating a separate processor to each subband particle filter, the overall computation time to track multiple objects in video could be significantly reduced.
4. Develop stand-alone visual tracking hardware by implementing our novel multi-scale trackers using Field Programmable Gate Arrays (FPGA), or Digital Signal Processor (DSP), boards.

References

[1] E. Maggio and A. Cavallaro, *Video tracking: theory and practice*: John Wiley & Sons, 2011.

[2] B. Zhang, Z. Li, A. Perina, A. Del Bue, V. Murino, and J. Liu, "Adaptive local movement modeling for robust object tracking," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 27, pp. 1515-1526, 2017.

[3] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing,* vol. 74, pp. 3823-3831, 2011.

[4] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR),* vol. 38, p. 13, 2006.

[5] B. Deori and D. M. Thounaojam, "A survey on moving object tracking in video," *International Journal on Information Theory (IJIT),* vol. 3, 2014.

[6] G. M. Rao and C. Satyanarayana, "Visual object target tracking using particle filter: a survey," *International Journal of Image, Graphics and Signal Processing,* vol. 5, p. 1250, 2013.

[7] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 2, pp. 2970-2979, 2014.

[8] J. J. Athanesious and P. Suresh, "Systematic survey on object tracking methods in video," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* vol. 1, pp. pp: 242-247, 2012.

[9] J. R. Raol, *Data Fusion Mathematics: Theory and Practice*: CRC Press, 2015.

[10] N. Zheng and J. Xue, *Statistical learning and pattern analysis for image and video processing*: Springer Science & Business Media, 2009.

[11] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 25, pp. 564-577, 2003.

[12] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision,* vol. 26, pp. 63-84, 1998.

[13] S. Avidan, "Support Vector Tracking " *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* vol. 1, 2001.

[14] M. Isard and A. Blake, "Condensation–Conditional Density Propagation for Visual Tracking,(1998)," *International Journal of Computer Vision Publ,* pp. 5-28.

[15] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*: Artech House, 2013.

[16] I. Leang, S. Herbin, B. Girard, and J. Droulez, "On-line fusion of trackers for single-object tracking," *Pattern Recognition*, vol. 74, pp. 459-473, 2018.

[17] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, pp. 174-188, 2002.

[18] W. L. Dunn and J. K. Shultis, *Exploring Monte Carlo Methods*: Elsevier, 2011.

[19] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*: Springer Science & Business Media, 2006.

[20] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo methods in practice*, ed: Springer, 2001, pp. 3-14.

[21] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE transactions on pattern analysis and machine intelligence*, pp. 90-99, 1986.

[22] R. Rosales and S. Sclaroff, "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, pp. 117-123.

[23] S. Mills, T. P. Pridmore, and M. Hills, "Tracking in a Hough Space with the Extended Kalman Filter," in *BMVC*, 2003, pp. 1-10.

[24] P. V. Hough, "Method and means for recognizing complex patterns," 1962.

[25] V. Lippiello, B. Siciliano, and L. Villani, "Visual motion estimation of 3D objects: an adaptive extended Kalman filter approach," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004, pp. 957-962.

[26] D. Beymer and K. Konolige, "Real-time tracking of multiple people using continuous detection," in *IEEE Frame Rate Workshop*, 1999, pp. 1-8.

[27] N. Li, L. Liu, and D. Xu, "Corner feature based object tracking using Adaptive Kalman Filter," in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, 2008, pp. 1432-1435.

[28] Q. Chen, Q.-S. Sun, P. A. Heng, and D.-S. Xia, "Two-stage object tracking method based on kernel and active contour," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, pp. 605-609, 2010.

[29] Q. Miao, G. Wang, X. Lin, Y. Wang, C. Shi, and C. Liao, "Scale and rotation invariant feature-based object tracking via modified on-line boosting," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 3929-3932.

[30] X. Wu, X. Mao, L. Chen, and A. Compare, "Combined motion and region-based 3D tracking in active depth image sequence," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, 2013, pp. 1734-1739.

[31] K. Narsimlu, T. R. Kanth, and D. R. Guntupalli, "Autonomous visual tracking with extended Kalman filter estimator for micro aerial vehicles," in *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015)*, 2015, pp. 31-42.

[32] M. Felsberg and F. Larsson, "Learning Bayesian tracking for motion estimation," in *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis-MLVMA'08*, 2008.

[33] M. Felsberg and G. Granlund, "Fusing dynamic percepts and symbols in cognitive systems," in *International Conference on Cognitive Systems*, 2008.

[34] X. Liu, Z. Lin, and S. T. Acton, "A grid-based Bayesian approach to robust visual tracking," *Digital Signal Processing*, vol. 22, pp. 54-65, 2012.

[35] Q. Sang, Z. Lin, and S. T. Acton, "A grid-based tracker for erratic targets," *Pattern Recognition*, vol. 48, pp. 3527-3541, 2015.

[36] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, "A color-based particle filter for multiple object tracking in an outdoor environment," *Artificial Life and Robotics*, vol. 15, pp. 41-47, 2010.

[37] C. Liu, P. Liu, J. Liu, J. Huang, and X. Tang, "2D articulated pose tracking using particle filter with partitioned sampling and model constraints," *Journal of Intelligent and Robotic Systems*, vol. 58, pp. 109-124, 2010.

[38] S. L. Tang, Z. Kadim, K. M. Liang, and M. K. Lim, "Hybrid blob and particle filter tracking approach for robust object tracking," *Procedia Computer Science*, vol. 1, pp. 2549-2557, 2010.

[39] I.-C. Chang and S.-Y. Lin, "3D human motion tracking based on a progressive particle filter," *Pattern Recognition*, vol. 43, pp. 3621-3635, 2010.

[40] X. Wang and Z. Tang, "Modified particle filter-based infrared pedestrian tracking," *Infrared Physics & Technology*, vol. 53, pp. 280-287, 2010.

[41] J. Zhao and Z. Li, "Particle filter based on Particle Swarm Optimization resampling for vision tracking," *Expert Systems with Applications*, vol. 37, pp. 8910-8914, 2010.

[42] R. C. Eberhart, Y. Shi, and J. Kennedy, "Swarm Intelligence (The Morgan Kaufmann Series in Evolutionary Computation)," 2001.

[43] L. Jing and P. Vadakkepat, "Interacting MCMC particle filter for tracking maneuvering target," *Digital Signal Processing*, vol. 20, pp. 561-574, 2010.

[44] B. Pu, F. Zhou, and X. Bai, "Particle filter based on color feature with contour information adaptively integrated for object tracking," in *2011 Fourth International Symposium on Computational Intelligence and Design*, pp. 359-362.

[45] X. Lu, L. Song, S. Yu, and N. Ling, "Object contour tracking using multi-feature fusion based particle filter," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, 2012, pp. 237-242.

[46] S. Fazli, H. M. Pour, and H. Bouzari, "Particle filter based object tracking with sift and color feature," in *Machine Vision, 2009. ICMV'09. Second International Conference on*, 2009, pp. 89-93.

[47] K. Hossain, C.-m. Oh, C.-W. Lee, and G.-S. Lee, "Multi-Part SIFT feature based particle filter for rotating object tracking," in *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*, 2012, pp. 1016-1020.

[48] H.-Y. Shen, S.-F. Sun, X.-B. Ma, Y.-C. Xu, and B.-J. Lei, "Comparative study of color feature for particle filter based object tracking," in *Machine Learning and Cybernetics (ICMLC), 2012 International Conference on*, pp. 1104-1110.

[49] H.-B. Kim and K.-B. Sim, "A particular object tracking in an environment of multiple moving objects," in *Control Automation and Systems (ICCAS), 2010 International Conference on*, 2010, pp. 1053-1056.

[50] D. Varas and F. Marques, "A region-based particle filter for generic object tracking and segmentation," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 1333-1336.

[51] Z. H. Khan, I. Y.-H. Gu, and A. G. Backhouse, "A robust particle filter-based method for tracking single visual object through complex scenes using dynamical object shape and appearance similarity," *Journal of Signal Processing Systems*, vol. 65, pp. 63-79, 2011.

[52] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, pp. 142-149.

[53] J. Martínez-del-Rincón, C. Orrite, and C. Medrano, "Rao–Blackwellised particle filter for colour-based tracking," *Pattern Recognition Letters*, vol. 32, pp. 210-220, 2011.

[54] P. Li and F. Chaumette, "Image cues fusion for object tracking based on particle filter," *Articulated Motion and Deformable Objects*, pp. 99-110, 2004.

[55] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *European Conference on Computer Vision*, 2002, pp. 67-81.

[56] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras, "Integration of conditionally dependent object features for robust figure/background segmentation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005, pp. 1713-1720.

[57] M. Z. Islam, C.-M. Oh, and C. W. Lee, "An efficient multiple cues synthesis for human tracking using a particle filtering framework," *International Journal of Innovative Computing, Information and Control*, vol. 7, pp. 3379-3393, 2011.

[58] P. Vadakkepat and L. Jing, "Improved particle filter in sensor fusion for tracking randomly moving object," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, pp. 1823-1832, 2006.

[59] T. A. Biresaw, A. Cavallaro, and C. S. Regazzoni, "Tracker-level fusion for robust bayesian visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, pp. 776-789, 2015.

[60] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, pp. 674-693, 1989.

[61] R. Gonzalez and R. Woods, "Digital image processing: Pearson prentice hall," *Upper Saddle River, NJ*, 2008.

[62] L. Kaur, S. Gupta, and R. Chauhan, "Image Denoising Using Wavelet Thresholding," in *ICVGIP*, 2002, pp. 16-18.

[63] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE transactions on image processing*, vol. 9, pp. 1532-1546, 2000.

[64] S. Arivazhagan and L. Ganesan, "Texture classification using wavelet transform," *Pattern recognition letters*, vol. 24, pp. 1513-1521, 2003.

[65] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse image and signal processing: wavelets, curvelets, morphological diversity*: Cambridge university press, 2010.

[66] S. Mallat, *A wavelet tour of signal processing: the sparse way*: Academic press, 2008.

[67] C. K. Chui, *An introduction to wavelets*: Elsevier, 2016.

[68] F. Jin, *Wavelet-based image and video processing*: University of Waterloo, 2004.

[69] J. Gomes and L. Velho, *From fourier analysis to wavelets* vol. 3: Springer, 2015.

[70] M.-Y. Shih and D.-C. Tseng, "A wavelet-based multiresolution edge detection and tracking," *Image and Vision Computing*, vol. 23, pp. 441-451, 2005.

[71] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells, "Noise reduction using an undecimated discrete wavelet transform," *IEEE Signal Processing Letters,* vol. 3, pp. 10-12, 1996.

[72] J.-L. Starck, "Nonlinear multiscale transforms," in *Multiscale and Multiresolution Methods*, ed: Springer, 2002, pp. 239-278.

[73] O. Prakash and A. Khare, "Tracking of moving object using energy of biorthogonal wavelet transform," *Chiang Mai J. Sci.*, vol. 42, pp. 783-795, 2015.

[74] F.-H. Cheng and Y.-L. Chen, "Real time multiple objects tracking and identification based on discrete wavelet transform," *Pattern Recognition,* vol. 39, pp. 1126-1139, 2006.

[75] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, "Tracking of moving objects by using a low resolution image," in *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on*, 2007, pp. 408-408.

[76] C.-H. Hsia, J.-M. Guo, and J.-S. Chiang, "Improved low-complexity algorithm for 2-D integer lifting-based discrete wavelet transform using symmetric mask-based scheme," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 19, pp. 1202-1208, 2009.

[77] T. G. Allen, M. R. Luettgen, and A. S. Willsky, "Multiscale appoaches to moving target detection in image sequences," *Optical Engineering,* vol. 33, pp. 2248-2254, 1994.

[78] S. H. Shaikh, K. Saeed, and N. Chaki, "Moving object detection approaches, challenges and object tracking," in *Moving Object Detection Using Background Subtraction*, ed: Springer, 2014, pp. 5-14.

[79] G. Hua and Y. Wu, "Multi-scale visual tracking by sequential belief propagation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. I-I.

[80] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Computer vision—ECCV 2002*, pp. 661-675, 2002.

[81] S. D. Roy, S. D. Tran, L. S. Davis, and B. S. Vikram, "Multi-resolution tracking in space and time," in *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, 2008, pp. 352-358.

[82] E. Maggio, F. Smerladi, and A. Cavallaro, "Adaptive multifeature tracking in a particle filtering framework," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 17, pp. 1348-1359, 2007.

[83] M. Hu, Z. Liu, J. Zhang, and G. Zhang, "Robust object tracking via multi-cue fusion," *Signal Processing,* vol. 139, pp. 86-95, 2017.

[84] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part V. Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 41, pp. 1255-1321, 2005.

[85] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *European Conference on Computer Vision*, 2014, pp. 170-185.

[86] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1195-1202.

[87] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, 2012, pp. 425-432.

[88] L. Cehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 941-953, 2013.

[89] J. Xue and N. Zheng, "Robust tracking with and beyond visible spectrum: a four-layer data fusion framework," in *Advances in Machine Vision, Image Processing, and Pattern Analysis*, ed: Springer, 2006, pp. 1-16.

[90] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 1838-1845.

[91] Y. Mroueh, T. Poggio, L. Rosasco, and J.-J. Slotine, "Multiclass learning with simplex coding," in *Advances in Neural Information Processing Systems*, 2012, pp. 2789-2797.

[92] S. Avidan, "Ensemble tracking," *IEEE transactions on pattern analysis and machine intelligence,* vol. 29, 2007.

[93] X. Li, C. Shen, A. Dick, and A. Van Den Hengel, "Learning compact binary codes for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2419-2426.

[94] S. Avidan, "Support vector tracking," *IEEE transactions on pattern analysis and machine intelligence,* vol. 26, pp. 1064-1072, 2004.

[95] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, pp. 1527-1554, 2006.

[96] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, pp. 504-507, 2006.

[97] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Transactions on Image Processing*, vol. 24, pp. 1424-1435, 2015.

[98] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in neural information processing systems*, 2013, pp. 809-817.

[99] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[100] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293-4302.

[101] H. Fan and H. Ling, "Sanet: Structure-aware network for visual tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 2217-2224.

[102] J. Choi, J. Kwon, and K. M. Lee, "Visual tracking by reinforced decision making," *arXiv preprint arXiv:1702.06291*, 2017.

[103] H. Hassanpour, M. Sedighi, and A. R. Manashty, "Video frame's background modeling: Reviewing the techniques," *Journal of Signal and Information Processing*, vol. 2, p. 72, 2011.

[104] P. Prasad, M. Kumar, and G. S. B. Rao, "Design of Biorthogonal Wavelets Based on Parameterized Filter for the Analysis of X-ray Images," in *Computational Intelligence in Data Mining-Volume 2*, ed: Springer, 2015, pp. 99-110.

[105] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 74-90, 1987.

[106] Z. He, *Wavelet Analysis and Transient Signal Processing Applications for Power Systems*: John Wiley & Sons, 2016.

[107] P. Patidar, M. Gupta, S. Srivastava, and A. K. Nagawat, "Image de-noising by various filters for different noise," *International journal of computer applications*, vol. 9, 2010.

[108] E. S. A. Ahmed, R. E. Elatif, and Z. T. Alser, "Median filter performance based on different window sizes for salt and pepper noise removal in gray and RGB images," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, pp. 343-352, 2015.

[109] C.-H. Hsia, J.-S. Chiang, and J.-M. Guo, *Multiple Moving Objects Detection and Tracking Using Discrete Wavelet Transform*: INTECH Open Access Publisher, 2011.

[110] D. Rowe, I. Huerta, J. Gonzàlez, and J. J. Villanueva, "Robust multiple-people tracking using colour-based particle filters," in *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 113-120.

[111] J. J. Pantrigo, J. Hernández, and A. Sánchez, "Multiple and variable target visual tracking for video-surveillance applications," *Pattern Recognition Letters,* vol. 31, pp. 1577-1590, 2010.

[112] P. Perez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE,* vol. 92, pp. 495-513, 2004.

[113] C. X. Tianzhu Zhang, Ming-Hsuan Yang "Learning Multi-task Correlation Particle Filters for Visual Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)* 23 January 2018.

[114] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 2544-2550.

[115] M. Dai, S. Cheng, X. He, and D. Wang, "A Structural Correlation Filter Combined with A Multi-task Gaussian Particle Filter for Visual Tracking," *arXiv preprint arXiv:1803.05845,* 2018.

[116] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 58-66.

[117] J. M. Mendel, *Lessons in digital estimation theory*: Prentice-Hall, Inc., 1986.

[118] N. Kingsbury, "The dual-tree complex wavelet transform: a new efficient tool for image restoration and enhancement," in *Signal Processing Conference (EUSIPCO 1998), 9th European*, 1998, pp. 1-4.

[119] F. Jin, "Wavelet-based image and video processing," University of Waterloo, 2004.

[120] R. Singh, R. K. Purwar, and N. Rajpal, "A better approach for object tracking using dual-tree complex wavelet transform," in *Image Information Processing (ICIIP), 2011 International Conference on*, 2011, pp. 1-5.

[121] R. Coifman, G. Matviyenko, and Y. Meyer, "Modulated Malvar–Wilson bases," *Applied and Computational Harmonic Analysis,* vol. 4, pp. 58-61, 1997.

[122] D. Moshe, "Denoising using wavelets."

[123] P. Arora and M. Bansal, "Comparative analysis of advanced thresholding methods for Speech-Signal denoising," *International Journal of Computer Applications,* vol. 59, 2012.

[124] D. L. Donoho and I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data," in *Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE*, 1994, pp. A24-A25 vol. 1.

[125] M. Wickerhauser, "2] RR Coifman, Y. Meyer, and MV Wickerhauser, Wavelet analysis and signal pro-cessing, Wavelets and Their Applications, ed. Ruskai et al., ISBN 0-86720-225-4, Jones and Bartlett, Boston, 1992, pp. 153 178," *IEEE Transactions on Information Theory*, vol. 32, pp. 712-718, 1992.

[126] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on information theory*, vol. 38, pp. 713-718, 1992.

[127] D. Wang, D. Miao, and C. Xie, "Best basis-based wavelet packet entropy feature extraction and hierarchical EEG classification for epileptic detection," *Expert Systems with Applications*, vol. 38, pp. 14314-14320, 2011.

APPENDIX A: Multiresolution image processing

The multi-resolution approach to image (or signal) processing and analysis is also known as MRA analysis. MRA aims to represent and analyze a signal (or image) at different resolution levels, such that features that might be ambiguous in one level may become evident in another. MRA approach unifies techniques from various disciplines including 1) pyramid image processing, 2) subband coding (filter banks) from signal processing, and 3) wavelet transforms. This appendix describes these different multi-resolution approaches.

1. Image pyramids

An image pyramid is a powerful and straightforward technique for representing an image at different resolution levels. It has been used in many applications including machine vision and image compression. In an image pyramid representation, a collection of decreasing image resolution forms a pyramid structure as shown in Figure 1. The base of the pyramid contains a high-resolution representation of the image, while the apex of

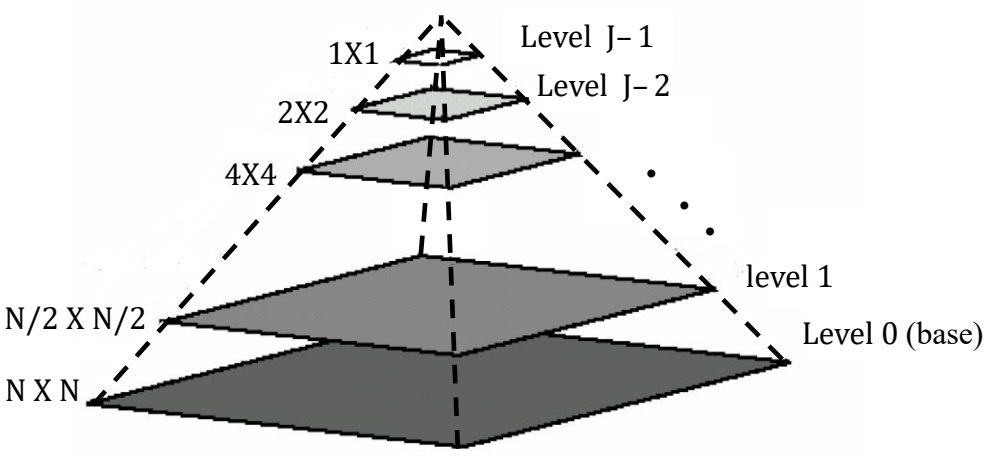


Figure 1. A pyramidal image structure

the pyramid includes a low-resolution approximation. The resolution, thereby, the size of the image, decreases while moving up the pyramid structure.

The size of base level is N X N or 2^J X 2^J, where $J = \log_2 N$, an intermediate level j is of size $2^{(J-j)}$ X$2^{(J-j)}$, where $0 \leq j \leq J$. A full pyramid structure composed of $J + 1$ resolution level whose size start from 2^J $X2^J$ to 2^0 X2^0. Usually, pyramids are truncated to $P + 1$ levels such that $J - P \leq j \leq J$. Therefore, the total number of elements in $P + 1$ pyramid levels is:

$$N^2 \left(1 + \frac{1}{(4)^1} + \frac{1}{(4)^2} + \cdots + \frac{1}{(4)^P}\right) \leq \frac{4}{3} N^2 \qquad (1)$$

Figure 2. shows a block diagram for generating image pyramids. The approximate image at level j is used to obtain the next one, i.e., level $j + 1$ approximation, in addition to level $j + 1$ prediction. We note that level $j + 1$ prediction represents the difference between the approximations at level j and level $j + 1$.

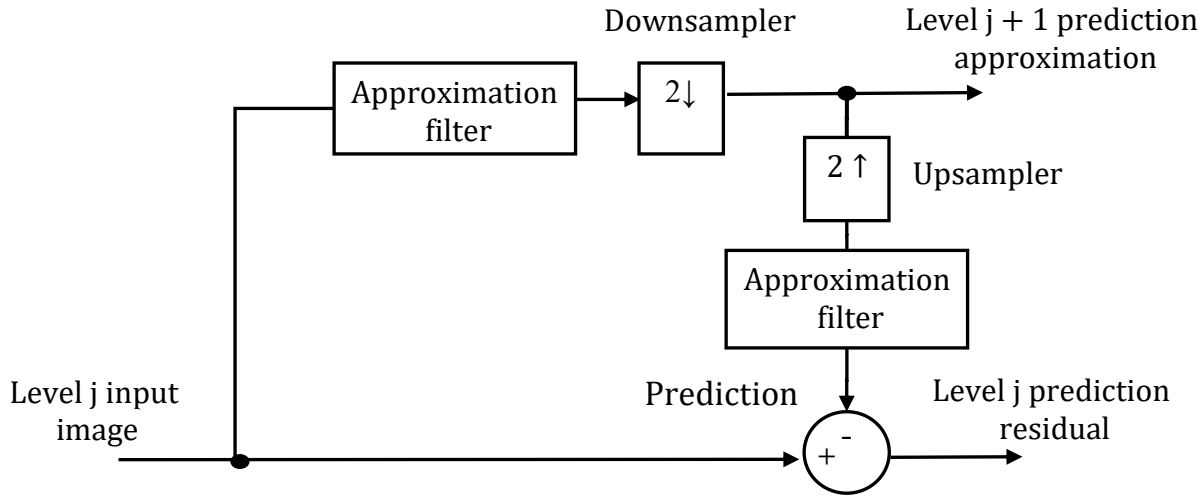


Figure 2. Block diagram for generating a pyramid representation

2. Subband coding

Another possible technique for multi-resolution analysis is *subband coding*. An image (or signal) is decomposed into a set of band-limited components, i.e., subbands that could be re-assembled to reconstruct the original image without error. Developed initially for speech and image compression, subbands are typically generated by bandpass filtering the input image (or signal). As the bandwidths of the resulting subbands are smaller than that of the original image, these subbands can be downsampled without any loss of information. To reconstruct the original image, one needs to sum an upsampled and filtered versions of the individual subbands.

Figure 3. represents a block diagram of a two-band subband coding and decoding system. The input is a one dimensional, band-limited discrete time signal $x(\mathrm{n})$ where $\mathrm{n} = 1, 2, \ldots$ and $\hat{x}(\mathrm{n})$ is generated by, first, the decomposition of $x(\mathrm{n})$ into $\mathrm{y}_0(\mathrm{n})$ and $\mathrm{y}_1(\mathrm{n})$

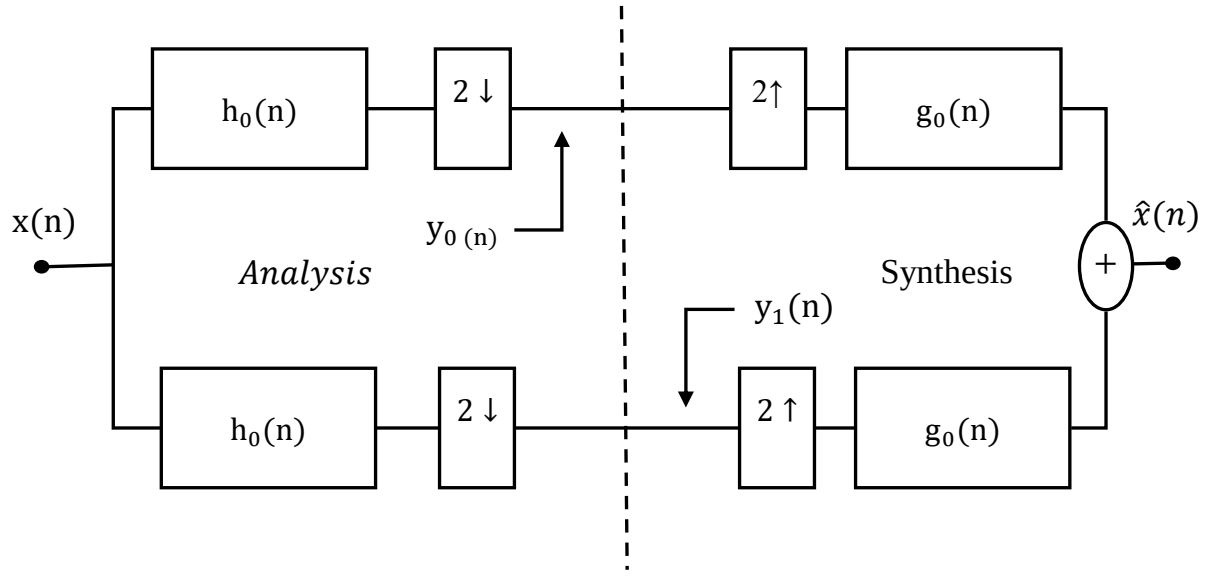


Figure 3. system block diagram of two band subband coding and decoding system

using the analysis filters, $h_0(n)$ and $h_1(n)$, and, second, by recombination using the synthesis filters $g_0(n)$ and $g_1(n)$.

3. Haar wavelet transform

The third technique for multi-resolution analysis is the Haar transform. This transform uses one of the oldest orthonormal basis functions on the real interval [0, 1], which was developed by Alfred Haar in 1909. The Haar wavelet transform is both separable and symmetric, and it could be expressed in matrix form as

$$\mathbf{T} = \mathbf{HFH} \tag{2}$$

where **F** is, e.g., an *N* x *N* image, then the transformation matrix **H** would also be an *N* x *N* matrix. The Haar wavelet transform could be viewed as one of the simplest orthonormal transforms, where its mother wavelet function $\psi(t)$ is given by:

$$\psi(t) = \begin{cases} 1 & 0 \le t \le \frac{1}{2}, \\ -1 & \frac{1}{2} \le t \le 1, \\ 0 & otherwise. \end{cases} \tag{3}$$

The scaling function $\varphi(t)$ of this Haar wavelet transform is given by:

$$\varphi(t) = \begin{cases} 1 & 0 \le t \le 1, \\ 0 & otherwise. \end{cases} \tag{4}$$

APPENDIX B: Review of probability distributions

Uncertainty about multiple random variables could be expressed by three types of probabilities including joint, marginal, and conditional probability distributions. This appendix aims to describe these probability distributions.

1. Joint probability

Given x and y be random variables which can take values in $X = \{v_1, v_2, \dots, v_m\}$, and $Y = \{w_1, w_2, \dots, w_n\}$, respectively. We can consider that (x, y)as a point in the vector *product space* of x and y. For each pair of the values $((v_i, w_j)$, where $1 \leq i \leq m$, and $1 \leq j \leq n$, we have *joint probability* $p_{ij} = \{x = v_i, y = w_j\}$. The value of p_{ij} is non-negative, also the sum of possible p_{ij} is equal to 1.

To obtain a full characterization of the pair of random variables (x, y), individually or together, we define *joint probability mass function* $P(x, y)$ in which:

$$P(x, y) \geq 0, \qquad and$$

$$\sum_{x \in X} \sum_{y \in Y} P(x, y) = 1 \tag{1}$$

From the *joint probability mass function* everything about x, and y can be computed. In particular, $P(x, y)$ could be used to get a separate *marginal distribution* for *x*, and *y* by summing over the other variable, i.e., the probability for one of the variables with no reference to any the other variables, as shown in the following equations:

$$P_X(x) = \sum_{y \in Y} P(x, y)$$

$$P_Y(y) = \sum_{x \in X} P(x, y) \tag{2}$$

For simplification, $P_X(x)$, and $P_Y(y)$ are usually referred to as $P(x)$, and $P(y)$ repectively.

2. Statistical independence

The random variables x and y are statistically independent, i.e., the probability of one variable is unaffected by the value of the other, if and only if

$$P(x, y) = P_X(x)\ P_Y(y) \tag{3}$$

3. Conditional probability

If the two variables *x*, and *y* are statistically dependent, then the value of one them yields to better estimate of the value of the other one. *Conditional probability* describes this statistical dependent such that

$$\Pr\{x = v_i | y = w_j\} = \frac{\Pr\{x = v_i, y = w_j\}}{\Pr\{y = w_j\}} \tag{4}$$

in terms of mass functions, *conditional probability* is expressed as

$$P(x|y) = \frac{P(x, y)}{P(y)} \tag{5}$$

In case of statistical independent of *x*, and *y*, this yields to $P(x|y) = P(x)$. That means that knowing the value y provides no information about x that was not already known from $P(x)$.