Multi-cue visual tracking: feature learning and fusion

Xiangyuan Lan

Follow this and additional works at: https://repository.hkbu.edu.hk/etd_oa

Recommended Citation

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at HKBU Institutional Repository. It has been accepted for inclusion in Open Access Theses and Dissertations by an authorized administrator of HKBU Institutional Repository. For more information, please contact repository@hkbu.edu.hk.
DATE: August 10, 2016

STUDENT'S NAME: LAN Xiangyuan

THESIS TITLE: Multi-cue Visual Tracking: Feature Learning and Fusion

This is to certify that the above student's thesis has been examined by the following panel members and has received full approval for acceptance in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairman: Prof. Ng Kwok Po
Chair Professor in Mathematics, Department of Mathematics, HKBU
(Designated by Dean of Faculty of Science)

Internal Members:
Dr. Choi Koon Kau
Associate Professor, Department of Computer Science, HKBU
(Designated by Head of Department of Computer Science)

Dr. Tam Hon Wah
Associate Professor, Department of Computer Science, HKBU

External Members:
Prof. Vijayakumar Bhagavatula
Professor
Department of Electrical and Computer Engineering
Carnegie Mellon University
USA

Prof. You Jia
Professor
Department of Computing
The Hong Kong Polytechnic University

In-attendance: Prof. Yuen Pong Chi
Professor, Department of Computer Science, HKBU

Issued by Graduate School, HKBU
Multi-cue Visual Tracking: Feature Learning and Fusion

LAN Xiangyuan

A thesis submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Principal Supervisor: Professor YUEN Pong Chi

Hong Kong Baptist University

August 2016
DECLARATION

I hereby declare that this thesis represents my own work which has been done after registration for the degree of PhD at Hong Kong Baptist University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

Signature: Lan Xiangyuan

Date: August 2016
Abstract

As an important and active research topic in computer vision community, visual tracking is a key component in many applications ranging from video surveillance and robotics to human computer interaction. Although it has been extensively studied in last two decades and significant progress has been made in recent years, it still remains challenging mainly due to numerous appearance variations caused by illumination, pose, occlusion, shape deformation and so on. Therefore, how to design an effective object appearance model which is able to handle different appearance changes has become one of the key problems in visual tracking. Since a single visual cue (feature) may not be sufficient to account for different appearance variations, multiple visual cues which describe different characteristics of the tracked object, e.g. color, texture, shape can be exploited jointly to achieve a more robust appearance model. In this thesis, we propose new appearance models based on multiple visual cues and address several research issues in feature learning and fusion for visual tracking.

Feature extraction and feature fusion are two key modules to construct the appearance model for the tracked target with multiple visual cues. Feature extraction aims to extract informative features for visual representation of the tracked target, and many kinds of hand-crafted feature descriptors which capture different types of visual information have been developed. However, since large appearance variations, e.g. occlusion, illumination may occur during tracking, the target samples may be contaminated/corrupted. As such, the extracted raw features may not be able to capture the intrinsic properties of the target appearance. Besides, without ex-
licitly imposing the discriminability, the extracted features may potentially suffer background distraction problem. To extract uncontaminated discriminative features from multiple visual cues, this thesis proposes a novel robust joint discriminative feature learning framework which is capable of 1) simultaneously and optimally removing corrupted features and learning reliable classifiers, and 2) exploiting the consistent and feature-specific discriminative information of multiple feature. In this way, the features and classifiers learned from potentially corrupted tracking samples can be better utilized for target representation and foreground/background discrimination.

As shown by the Data Processing Inequality, information fusion in feature level contains more information than that in classifier level. In addition, not all visual cues/features are reliable, and thereby combining all the features may not achieve a better tracking performance. As such, it is more reasonable to dynamically select and fuse multiple visual cues for visual tracking. Based on aforementioned considerations, this thesis proposes a novel joint sparse representation model in which feature selection, fusion, and representation are performed optimally in a unified framework. By taking advantages of sparse representation, unreliable features are detected and removed while reliable features are fused on feature level for target representation. In order to capture the non-linear similarity of features, the model is further extended to perform feature fusion in kernel space. Experimental results demonstrate the effectiveness of the proposed model.

Since different visual cues extracted from the same object should share some commonalities in their representations and each feature should also have some diversities to reflect its complementarity in appearance modeling, another important problem in feature fusion is how to learn the commonality and diversity in the fused representations of multiple visual cues to enhance the tracking accuracy. Different from existing multi-cue sparse trackers which only consider the commonalities among the sparsity patterns of multiple visual cues, this thesis proposes a novel multiple sparse representation model for multi-cue visual tracking which jointly exploits
the underlying commonalities and diversities of different visual cues by decomposing multiple sparsity patterns. Moreover, this thesis introduces a novel online multiple metric learning to efficiently and adaptively incorporate the appearance proximity constraint, which ensures that the learned commonalities of multiple visual cues are more representative. Experimental results on tracking benchmark videos and other challenging videos show that the proposed tracker achieves better performance than the existing sparsity-based trackers and other state-of-the-art trackers.

In short, the major contributions of this thesis are summarized as follows.

- A robust feature learning model is proposed to learn uncontaminated and discriminative features from multiple visual cues for feature extraction in visual tracking.

- A robust feature fusion model based on joint sparse representation is proposed to dynamically select and fuse multiple visual cues on feature level for visual tracking.

- A multiple sparse representation framework is proposed to explicitly model the commonality and diversity in the representation of multiple visual cues, which achieves more accurate appearance modeling with multiple features in visual tracking.

**Keywords:** visual tracking, feature learning, feature fusion
Acknowledgements

I would like to take this good opportunity to express my appreciation to all the people who have helped me during my doctoral study.

First of all, I would like to express my sincere gratitude to my principle supervisor Prof. Pong Chi Yuen for providing me the precious opportunity to pursue graduate studies in the challenging yet motivating areas of computer vision and mentoring throughout my graduate life with his patience, wisdom, and great effort. Prof. Yuen always give me maximum freedom to explore problems, and his timely constructive suggestions always help me out of perplexity and inspire me to attack more challenging problems and seek higher academic objectives. His guidance, encouragement and support also give me plenty of chances to expand my perspectives and gain knowledge. Working with Prof. Yuen is an enjoyable experience and will benefit me throughout my life for not only how to do research of high quality, but also how to be a man of humility, integrity, kindness and deep insight.

My deep gratitude also goes to Prof. Rama Chellappa for his precious advice and valuable support to my research. It is a great honor and a memorable experience for me to work with him in Computer Vision Laboratory of The University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA.

I would like to thank all the faculty members and staffs in the Department of Computer Science at Hong Kong Baptist University for their kind assistance and support to my study.

I also would like to thank all my friends and colleagues who helped me during
my study in Hong Kong and USA. Especially, I wish to express my thanks to Dr.
Jinhua Ma and Dr. Shengping Zhang for their cooperations in my research, as well
as Dr. Weiwen Zou, Dr. Meng-Hui Lim, Dr. Shu-Lim Ho, Dr. Na Liu, Dr. Haiying
Tian, Mr. Jiawei Li, Mr. Guangcan Mai, Miss. Baoyao Yang, Mr. Siqui Liu, Mr.
Kin Sun Chan, and Mr. Zexiong Cai for their discussions and help about my life
and research.

Finally, I would like to express my heartfelt gratitude to my parents and Miss
Wei Wang for their sincere love, encouragement and deep understanding all the time.
Table of Contents

Declaration i

Abstract ii

Acknowledgements v

Table of Contents vii

List of Tables xi

List of Figures xii

List of Abbreviations xv

Chapter 1 Introduction 1

1.1 Background and Motivation ........................................... 1
  1.1.1 Background ..................................................... 1
  1.1.2 Motivations of This Project ................................. 3

1.2 Review of Related Visual Tracking Methods ....................... 5
  1.2.1 Generative Trackers .......................................... 6
  1.2.2 Discriminative Trackers ..................................... 7
  1.2.3 Sparsity-based Trackers .................................... 8
  1.2.4 Metric Learning-based Trackers ......................... 10
  1.2.5 Feature Learning-based Trackers ......................... 10

1.3 Contributions of This Thesis .................................... 11
Chapter 2  Robust Joint Discriminative Feature Learning for Multi-Cue Visual Tracking  16

2.1 Introduction  ........................................... 16
2.2 Related Work  ........................................... 19
  2.2.1 Shared and Feature-Specific Information among Multiple Features/Modalities/Views for Pattern Classification  .................. 20
2.3 Proposed Model ........................................... 20
  2.3.1 Robust Joint Discriminative Feature Learning  ............... 20
  2.3.2 Optimization ........................................ 23
2.4 Implementation Details .................................... 27
  2.4.1 Target Representation ................................ 27
  2.4.2 Observation Likelihood for Particle Filtering .................. 27
2.5 Experiments .............................................. 28
  2.5.1 Experimental Setting .................................. 28
  2.5.2 Experimental Results .................................. 29
2.6 Conclusion ................................................ 32

Chapter 3  Joint Sparse Representation and Robust Feature-Level Fusion for Multi-Cue Visual Tracking  34

3.1 Introduction .............................................. 34
3.2 Related Work .............................................. 36
  3.2.1 Multi-Task Joint Sparse Representation ....................... 36
3.3 Robust Feature-Level Fusion for Multi-Cue Tracking  ............... 37
  3.3.1 Particle Filter ........................................ 37
  3.3.2 Feature-Level Fusion Based on Joint Sparse Representation .. 38
  3.3.3 Detecting Unreliable Visual Cues for Robust Fusion .......... 39
  3.3.4 Optimization Procedure ................................ 42
  3.3.5 Kernelized Framework for Robust Feature-Level Fusion ....... 45
### Chapter 3

#### 3.3.6 Computational Complexity

3.4 Implementation Details

- 3.4.1 Template Update Scheme
- 3.4.2 Tight Lipschitz Constant Estimation

#### 3.5 Experiments

- 3.5.1 Unreliable Feature Detection on Synthetic Data
- 3.5.2 Visual Tracking Experiments

#### 3.6 Conclusion

---

### Chapter 4

**Multiple Sparse Representations with Commonality and Diversity Modeling for Multi-Cue Visual Tracking**

#### 4.1 Introduction

#### 4.2 Related Work
- 4.2.1 Commonality and Diversity Modeling in Pattern Classification and Recognition

#### 4.3 Proposed Tracking Algorithm

- 4.3.1 Particle Filter
- 4.3.2 Fusion of Multiple Sparse Representations with the Proximity Constraint
- 4.3.3 Learning Adaptive Proximity Constraint Using Online LogDet Regularized Multiple Metric Learning
- 4.3.4 Optimization Procedure
- 4.3.5 Discussion

#### 4.4 Implementation Details

- 4.4.1 Model Update Scheme

#### 4.5 Experiments

- 4.5.1 Experiment Setting
- 4.5.2 Evaluation on Publicly Available Sequences
- 4.5.3 Evaluation on Visual Tracking Benchmark

#### 4.6 Conclusion
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>97</td>
</tr>
<tr>
<td>5.1</td>
<td>Summary</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>Future work</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Curriculum Vitae</td>
<td>120</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Center Location Error. The best three results are shown in red, green and blue. .................................................. 30
2.2 Success Rate. The best three results are shown in red, green and blue. 30
3.1 The Challenging Factor and Its Corresponding Videos . . . . . . . . . . 54
3.2 Center Location Error (in pixels). The best three results are shown in red, blue and green. ............................... 57
3.3 VOC Overlapping Rate. The best three results are shown in red, blue and green. ................................................. 60
3.4 Success Rate. The best three results are shown in red, blue and green. 62
4.1 Center Location Error. The best three results are shown in red, blue and green. .................................................. 88
4.2 Overlapping Rate. The best three results are shown in red, blue and green. 88
List of Figures

2.1 Quantitative frame-by-frame comparison of 11 trackers on 4 Challenging videos in terms of center location error ........................................ 31

2.2 Qualitative results on some typical frames including some challenging factors (video name). (a) Illumination (Car4, Car11, Trellis). (b) Pose (Shaking, Mountain-bike, Skating1). (c) Occlusion (Subway, DavidOutdoor, Faceocc2). (d) Cluttered background (Crossing, Animal, Bolt) .............................................................. 32

3.1 Graphical illustration of unreliable feature detection on synthetic data 52

3.2 Graphical Illustration of Unreliable Feature Detection Scheme. The upper-left image in purple box show the tracking result in the 658-th frame of video Occluded Face 1. The seven figures in the upper-right green box illustrate the resulting reconstruction coefficients via solving the $\ell_1$ minimization problem as shown in small green box. After solving (3.3.24), the resulting reconstruction coefficients of each feature encoded in matrix $R$ are illustrated in the bottom-left blue box, and the unreliable detection results encoded in matrix $S$ are illustrated in the bottom-right blue box. The $y$-coordinate of all above figures denotes the value of coefficient, and the $x$-coordinate denotes the index of the template in the template set. .......................................................... 53
3.3 The survival curves based on the F-score for the entire thirty-three videos and the subset of videos containing different challenging factors, and the average F-scores are shown in the corresponding legend in descending order. (a) Whole set of videos. (b) Occlusion subset of videos. (c) Cluttered background subset of videos. (d) Scale subset of videos. (e) Illumination subset of videos. (f) Pose subset of videos.

3.4 Qualitative results on some typical frames including some challenging factors. (a) Occlusion. (b) Background. (c) Scale. (d) Illumination. (d) Pose.

4.1 Graphical Illustration of Commonality and Diversity of Feature Representation. (a) A man whose face undergoes illumination and pose variation. (b) The sparse pattern for each visual cue including the HOG and the local covariance descriptors obtained by solving an $\ell_1$ norm minimization problem. (c) The diverse pattern. (d) The common pattern. The $y$-coordinate of all above figures denotes the value of the sparse patterns, and the $x$-coordinate denotes the index of the template in the template set.

4.2 Qualitative results on some typical frames including some challenging factors. (a) Illumination and cluttered background. (b) Deformation and occlusion.

4.3 The precision plots and the success plots of One-Pass-Evaluation on the subsets of sequence contains different challenging factors. (a) Fast motion and background clutter. (b) Motion blur and deformation. (c) In-plane and out-plane rotation. (d) Low resolution and occlusion. (e) Illumination and out of view. (f) Scale variation. The ranking result and performance score of each tracker is shown in the legend of each figure. The precision plots and the success plots of One-Pass-Evaluation of all video sequence are shown in the last two figures of (f).
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>The survival curve based on the F-score for the entire fifty-one videos.</td>
<td>95</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of the proposed methods</td>
<td>96</td>
</tr>
</tbody>
</table>
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMM</td>
<td>Alternating Direction Method of Multipliers</td>
</tr>
<tr>
<td>ALOV++</td>
<td>Amsterdam Library of Ordinary Videos data set</td>
</tr>
<tr>
<td>APG</td>
<td>Accelerated Proximal Gradient Method</td>
</tr>
<tr>
<td>ASLA</td>
<td>Adaptive Structural Local Appearance Model-based Tracker</td>
</tr>
<tr>
<td>CDMT</td>
<td>Multiple Sparse Representations with Commonality and Diversity Modeling for Multi-Cue Visual Tracking</td>
</tr>
<tr>
<td>COM</td>
<td>Commonality Modeling-based Tracker</td>
</tr>
<tr>
<td>CSK</td>
<td>Circulant Structure Tracker</td>
</tr>
<tr>
<td>CLE</td>
<td>Center Location Error</td>
</tr>
<tr>
<td>CT</td>
<td>Compressive Tracker</td>
</tr>
<tr>
<td>DFT</td>
<td>Distribution Fields-based Tracker</td>
</tr>
<tr>
<td>DPI</td>
<td>Data Processing Inequality</td>
</tr>
<tr>
<td>EP</td>
<td>Tracker with Proximity Constraint defined by Euclidean Distance</td>
</tr>
<tr>
<td>GLF</td>
<td>Gradient Logarithm Field</td>
</tr>
<tr>
<td>HOG</td>
<td>Histograms of Oriented Gradients</td>
</tr>
<tr>
<td>IVT</td>
<td>Incremental Learning-based Visual Tracker</td>
</tr>
</tbody>
</table>
JSRFFT Joint Sparse Representation based Feature-level Fusion Tracker
K-RJSRFFT Robust Joint Kernel Sparse Representation based Feature-Level Fusion Tracker
L1APG Accelerated Proximal Gradient-based $\ell_1$ Tracker
L1T $\ell_1$ Tracker
LEG0 Logdet Exact Gradient Online Metric Learning Algorithm
LPBoost Two-class Linear Programming Boosting
LogDet Log Determinant
LOT Locally Orderless Tracker
MIL Multiple Instance Learning-based Tracker
MKL Multiple Kernel Learning
MTJSR Multi-Task Joint Sparse Representation
MTMVT Multi-Task Multi-View Tracker
MTT Multi-Task Sparse Tracker
OAB Online AdaBoost-based Tracker
OPE One-Pass-Evaluation
PCA Principle Component Analysis
RJDFT Robust Joint Discriminative Feature Learning for Visual Tracking
RJSRFFT Robust Joint Sparse Representation-based Feature-Level Fusion Tracker
RPCA Robust Principle Component Analysis
VOC Visual Object Classes
SCM Sparse Collaborative Model-based Tracker
SemiT Semi-supervised Online Boosting-based Tracker
STRUCK Structure Output Support Vector Machine-based Tracker
SVM Support Vector Machine
Chapter 1

Introduction

This chapter gives an introduction to multi-cue visual tracking from the aspects of feature learning and fusion. The research background and motivation is introduced in Section 1.1. Then Section 1.2 reviews some existing visual tracking methods which are related to the work presented in this thesis. The contribution of this thesis are reported in Section 1.3. Finally, The overview of this thesis is given in Section 1.4.

1.1 Background and Motivation

1.1.1 Background

With the great advance in computation ability of modern computers and the significant reduction in the cost of high quality video cameras, the development of intelligent automated video analysis systems which have broad and promising applications in robotics [18, 91], traffic monitoring [15, 105], video surveillance [38, 16], human-computer interaction [92], and etc., have received great interests in both academia and industry. In particular, one main focus of designing such kind of systems is to make computers capable of motion perception which generally consists of three key steps: detecting moving objects of interest, tracking such objects in consecutive video frames, and recognizing behaviors and events based on the object
tracks in the video. Since the estimated motion state (i.e. location, size, orientation, etc.) and the captured appearance of the interesting moving objects in consecutive frames generated by the visual trackers provide indispensable information for motion understanding, and the performance of these systems depends heavily on the reliability of the estimation performed by the visual trackers, visual tracking has become a fundamental and important task in intelligent motion perception.

In general, a typical visual tracking system mainly consists of four components: target initialization, appearance model, motion model and searching strategies. Target initialization can be performed manually or by pre-learned object detectors, which provides the visual trackers with some prior knowledge about the tracked target and is executed to initialize the visual trackers by using information of the initial motion state and the target appearance. Motion model is used to estimate the motion state (e.g. location, scale, etc.) of the tracked target. Some dynamical state estimation technique such as Kalman filters [60], particle filters [51] are usually applied to build up the motion model. Searching strategies can be done by maximizing observation likelihood or classification confidence score based on the motion estimation. An appearance model generally consists of two key steps: feature extraction and feature representation. Feature extraction first extracts informative features from the tracked target for visual description, and then feature representation is performed by exploiting some statistical models for object representation and identification based on the extracted features.

The appearance of tracked targets may be changed dramatically during the tracking process due to some complicated factors (e.g. illumination variation, fast motion blur, occlusion, pose variation, etc.), which may cause tracking degradations and even failure. To deal with these challenging factors, how to develop an effective appearance model has become a key issue for the success of a visual tracker [72], and much effort have been done to enhance the tracking performance in terms of feature extraction and representation, which are the key steps in appearance modeling. For example, Zhou et al. [163] proposed to exploit the SIFT features to deal with
rotation and scale changes of target appearance. To handle dramatic illumination change, He et al. [39] proposed to extract the locality sensitive histograms to enhance the tracking robustness to illumination changes. The local covariance descriptors are exploited to capture the local correlation of pixels [49]. To increase the accuracy in target representation and identification based on the extract features, many statistical models which can be roughly categorized into generative and discriminative approach have been proposed for feature representation, such as subspace learning [93], sparse representation [87], boosting learning [33], SVM learning [3], etc..

Although many visual features with related statistical models have been proposed to account for different variations, due to the irregular changes in each individual appearance variation as well as their complicated interactions, exploiting a single features may not yield good tracking results. Therefore, a promising direction is to exploit multiple visual cues (features) jointly in both feature extraction and feature representation to deal with different kinds of variations in tracking. Along this direction, this thesis investigates and develops effective feature extraction strategies to obtain informative descriptions of the tracked target and reliable and accurate feature fusion schemes for target representation and identification so as to construct a robust multi-cue appearance model.

1.1.2 Motivations of This Project

Various hand-crafted feature extractors such as SIFT [82], HOG [20] have been developed to account for appearance variations (e.g. scale, illumination, etc.) in object detection and recognition. As result, a straightforward approach for feature extraction in multi-cue appearance modeling is directly employing these features generated from several kinds of feature extractors [33, 44, 33, 34, 4]. However, using the raw feature descriptors for appearance model may not accurately characterize the target appearance or achieve satisfactory tracking results. This is because: 1) The extracted raw features may be contaminated/corrupted due to some large exter-
nal variations (e.g. occlusion, illumination) and cannot well characterize the tracked object (e.g., color histogram under large illumination change), and 2) The extracted raw features may suffer the loss of discriminability without explicitly imposing the discriminative information into the feature extraction process, and thereby may not be well adapted to foreground/background discrimination for visual tracking.

To address these issues, feature learning should be performed to remove the contaminated/corrupted features and impose discriminability for feature extraction. Although several feature learning-based visual trackers have been proposed recently such as dictionary-based and deep learning-based approaches [117, 133, 83, 118, 68], most of them are developed under the context of using a single feature/modality which do not explicitly consider the complementarity of different features and may not be effective to be employed in feature fusion. Based on aforementioned considerations, this thesis proposes a novel feature learning framework to learn and extract uncontaminated discriminative features from multiple visual cues for appearance modeling.

With the extracted features from multiple visual cues, fusion can be performed on two levels: score level and feature level. It is generally agreed that feature-level fusion contains more information than score-level fusion, which can also be concluded form the Data Processing Inequality [19]. In addition, not all features are reliable, which implies that combining all features may not enhance the tracking performance. Although two feature level fusion methods, e.g. feature concatenation-based method [127] and multiple kernel learning (MKL)-based method [132], have been proposed, they may not achieve reliable tracking results because 1) feature concatenation ignores the incompatibility of different visual cues, and 2) pre-learned classifier based on MKL may be affected by corrupted features. To overcome the limitations of existing works and address these two issues in feature fusion, this thesis proposes a novel feature fusion-based representation model for visual tracking in which unreliable features are removed and reliable features are fused to represent the tracked target within a unified framework based on joint sparse representation.
To exploit the non-linear similarity between features which has been shown to able to enhance the performance in classification and recognition [135, 98, 28], this thesis also develops a general tracking algorithm which is able to dynamically select and fuse features in multiple kernel spaces.

Many research works (e.g. [137] [79] [47]) show that explicitly considering the commonality and diversity of different features can further enhance the performance of recognition and classification in many pattern classification tasks because commonality is closely related to the concept of agreement and complementarity is related to diversity, which suggests us a strategy to enhance the feature fusion performance in visual tracking. Considering that different features extracted from the same object should share some commonalities in their representations while each feature should also have some diversities to reflect its complementarity in appearance modeling, different from existing sparse representation-based multi-cue visual tracking algorithms which consider commonality only and may not well exploit the complementarity of different features [127, 48, 44], the thesis proposes a novel multiple sparse representation framework for visual tracking which simultaneously model commonality and diversities of the sparse patterns of different features. To ensure the representative object templates are activated for object representation, this thesis further incorporates the proximity information into the sparse representation framework by using a novel online multiple metric learning.

1.2 Review of Related Visual Tracking Methods

This section gives an overview of some related works in visual tracking. We first reviews some typical tracking methods from two categories: generative trackers and discriminative trackers, respectively. Because the proposed visual tracking models in this thesis are based on sparse representation and metric learning which addresses some research issues in feature learning and fusion, this section also discusses some related tracking algorithms based on sparse representation and metric learning as well as some trackers which address issues in feature learning and fusion, particularly.
For a more comprehensive review of tracking methods, the interested readers can refer to [116, 139, 94, 155, 72, 100, 136, 101, 76, 129].

1.2.1 Generative Trackers

The core idea of generative trackers is to construct an appearance model, which can be used to search for the tracked target with the lowest matching error. In the last two decades, much effort has been made to develop object appearance models, which include kernel-based methods [17, 67, 97], mixture models [53, 164], subspace learning [9, 93, 49, 123, 62, 102], linear representation [61, 88, 73, 157, 156, 154, 144], and so on. Japson et al. [53] constructed an online adaptive appearance model based on mixture models via Expectation-Maximization algorithm to handle appearance variations and occlusion during tracking. Comaniciu et al. [17] developed a mean-shift tracking framework in which the object represented by color histogram is located with mode shifts. Based on the mean-shift tracking framework [17], several variants of kernel-based trackers have been proposed, e.g. [97], [67]. However, most kernel-based trackers describe the object’s appearance based on holistic non-parametric distributions of features and hence cannot handle partial occlusion well. To alleviate this issue, Adam et al. [1] developed a patch-based appearance model, which also explicitly models the spatial relationship among patches. Inspired by some research findings in eigen subspace representation for objects’ appearance modeling [107], Black and Jepson [9] proposed to employ view-based eigen space representation for tracking within an optical flow framework. Instead of using pre-learned appearance models as in [9] which may not well account for large appearance changes, online subspace learning methods, e.g. [93, 49, 50, 160, 102] are exploited to learn adaptive and low-dimensional subspaces for appearance modeling. To enhance the robustness to combinatorial appearance changes, Kwon and Lee [62] decomposed the observation model into multiple basic models which account for different kinds of variations and motions. Linear representation-based approaches directly use linear combination of object templates to represent the object without learning subspace bases. In sparse
representation-based trackers, e.g. [87, 88], trivial templates are also introduced to model occlusion and image noises. To avoid solving the computationally expensive $\ell_1$ minimization problem in sparse representation, Li et al. [73] proposed a metric-weighted linear representation model, which is also able to capture the correlations of feature dimensions. Wang et al. [110] proposed the probability continuous outlier model, which exploits spatial information among outliers to detect and remove outliers. Recently, Siena et al. [99] introduce color profile-based method to detect occlusion which prevents inappropriate tracker updating.

1.2.2 Discriminative Trackers

Discriminative trackers regard tracking as a foreground/background classification problem, which aims to maximize the interclass separation so that the object can be distinguished from the background. Research works on discriminative trackers include SVM-based approaches [3, 71, 146], structure learning [36, 138, 151, 13], online boosting [33, 34, 135], multiple instance learning [5, 4, 147], compressive tracking [149, 150, 165], metric learning-based methods [46, 56, 57, 55], segmentation-based methods [121, 45, 134, 124, 30, 103, 24, 104], Gaussian process regression [27], correlation filter-based approaches [10, 22, 21, 41, 74, 148, 75, 85, 80, 80, 43], deep learning-based approaches [118, 113, 114, 68, 84], and so on. Early research work such as [3] learned SVM for tracking with off-line samples, which may not be able to handle unpredicted appearance variations. Therefore, online adaptive learning-based approaches are more favored for discriminative tracking. Online boosting-based methods such as [33, 34] are exploited for online feature selection in visual tracking. However, such methods may update the tracker with potential misaligned samples, which may lead to the tracking drift. To alleviate this problem, multiple instance learning-based methods [4, 147] were developed which trains tracking model with bag of samples. To alleviate labeling ambiguity caused by binary labels, online structured output SVM was introduced to incorporate the structural constraints which exploit the spatial relationship of samples for model learning. Corre-
lation filter-based trackers [10, 41] alleviate the sampling ambiguity with Gaussian-weighted labels. By generating virtual samples using cyclic shifts, they can be learned in the frequency domain efficiently. Recently, Zhang et al. [146] proposed a multi-expert tracking framework in which the best experts are selected based on minimum entropy criterion. Wu et al. [131] exploited the landmark-based label propagation which takes full advantages of the geometric structure of tracking samples for tracking.

1.2.3 Sparsity-based Trackers

Sparse Trackers with Single Feature Based on the intuition that the appearance of a tracked object can be sparsely represented by its appearances in previous frames, Mei and Ling [87] proposed the $\ell_1$ tracker which is robust to occlusion and noise. Subsequently, many research works have been done to improve the $\ell_1$ tracker in terms of the accuracy and efficiency [155]. Li et al. [69] exploited compressive sensing theory to improve the computational efficiency by reducing the template dimension. Liu et al. [78] developed a two-stage sparse optimization-based tracker in which sparse discriminative features are selected and temporal information is exploited for target representation via dynamic group sparsity algorithm. Liu et al. [77] proposed a mean-shift tracker based on histogram of local sparse code. Mei et al. [88] adopted $\ell_2$ minimization to bound the $\ell_1$ error in order to speed up particle resampling, and Bao et al. [7] utilized the accelerated proximal gradient method and introduced an $\ell_2$ norm regularization step on trivial template coefficients to speed up and improve the performance of the $\ell_1$ tracker. Zhong et al. [162] combined a holistic sparsity-based discriminative classifier and a local sparse generative model to handle occlusion, cluttered background and model drift. Zhang et al. [158] proposed a multi-task joint sparse learning method to exploit the relationship between particles such that the accuracy of $\ell_1$ tracker can be improved. Xu et al. [54] developed a local sparse appearance model to enhance robustness to occlusion. Wang et al. [112] introduced sparsity regularization to incremental subspace learning to account for
the noise caused by occlusion. Zhuang et al. [167] constructed a discriminative similarity map for reverse sparse-based tracking. In order to exploit a visual prior for tracking, Wang et al. [120] constructed a codebook from SIFT descriptors learned from a general image data set for sparse coding, and a linear classifier was trained on sparse codes for foreground/background classification. The trackers mentioned above utilized a single cue/feature for appearance modeling. To fuse multiple features, Wu et al. [127] concatenated multiple features into a high-dimensional feature vector to construct a template set for sparse representation. However, the high dimensionality of the combined feature vector increases the computational complexity of this method. And, fusion via concatenation may not be able to improve performance when some source data are corrupted. Wang et al. [115] introduced a kernel sparse representation-based tracking algorithm to fuse multiple features in kernel space for tracking, but this method assumes all features are reliable and it could not adaptively select reliable feature for tracking. Recently, Wu et al. [130] proposed metric learning-based structural appearance model in which weighted local sparse codes are fused by concatenation for appearance modeling.

Sparse Trackers with Multiple Features  Wu et al. [127] concatenated multiple features for designing a sparse representation of object appearance. Since the performance of feature concatenation may be downgraded if some corrupted features exist, they [127] employed a trivial template set to model the missing elements of corrupted features. However, trivial templates may be activated to represent any object, which leads to inaccurate object representation [7]. Inspired by multi-task joint sparse representation [142], Hu et al. [48] proposed a sparse tracker based on multi-feature joint sparse representation. Different from the feature concatenation-based sparse tracker [127], the feature fusion scheme in [48] did not require the representation coefficients of all feature templates to be the same. However, the assumption that all feature representations share the same sparsity pattern may not hold for tracking. Each feature should be allowed to have some diverse sparse patterns, which enables their complementarity to account for different object ap-
pearance variations. Hong et al. [44, 86] proposed a multi-task multi-view sparse tracker to fuse multiple features for appearance modeling and detect the outlier particles. This method assumes that all features of non-outlier particles share the same sparsity patterns.

1.2.4 Metric Learning-based Trackers

In order to enhance resistance to background distracters, Jiang et al. [56, 57] exploited neighborhood component analysis to learn adaptive metric for differential tracking. In [58], the adaptive metric for tracking is learned by unifying spatial and attribute selection. Wang et al. [122] exploited the principal metric learning framework to learn a discriminative metric for appearance modeling. Different from the aforementioned methods that learn the metric in a computationally expensive off-line manner, Li et al. [73] exploited online metric learning for metric-weighted least square regression-based tracking. The trackers mentioned above only learn a metric for a single feature. Although single feature-based metric learning algorithm can be applied to multiple features via feature concatenation, it ignores the different properties of multiple features and the learning efficiency may be degraded.

1.2.5 Feature Learning-based Trackers

Recent works on feature learning-based tracking include dictionary learning-based approaches and deep learning-based approaches. Various dictionary learning based-trackers are proposed to learn informative features for visual tracking. To perform effective model updating for the $\ell_1$ tracker, Wang et al. [117] developed an online nonnegative dictionary learning for template updating. To learn informative features which capture the intrinsic geometric structure of covariance feature descriptors for appearance modeling, Zhang et al. [152] proposed an online dictionary learning on the manifold of symmetric positive definite matrices. Yang et al. [133] employed the supervised incremental dictionary learning to enhance the reconstructive and discriminative power of the appearance model. Along this line, more vari-
nants of supervised dictionary learning-based trackers have been developed to exploit the relationship among particles [25], local information [83], etc.. Most dictionary learning-based trackers use a single feature, i.e. intensity only, which may not be sufficient to account for large appearance variations. Deep learning-based trackers such as [118, 68] tuned an off-line pre-trained deep neural network online to adapt the appearance variations, which may not be efficient. Different from the aforementioned approaches, this proposed feature learning model in this thesis aims to exploit multiple visual cues for feature learning without off-line large-scale training samples.

1.3 Contributions of This Thesis

This thesis addresses several important research issues in feature learning and fusion for multi-cue visual tracking. The major contributions of the thesis are summarized as follows:

1. A robust discriminative feature learning framework is proposed to extract discriminative uncontaminated features from track objects. Within this unified framework, by utilizing the representation and discriminative power of multiple visual cues, the contaminated features are removed and discriminability is imposed via learning reliable classifiers from multiple visual cues. In addition, a novel feature feature fusion scheme which considers the shared and feature-specific discriminative information from multiple cues is incorporated into the learning framework to enhance the fusion result in terms of the consistency and complementarity in discriminability of multiple visual cues. A four-step iterative optimization algorithm to effectively solve the proposed robust joint discriminative feature learning model. These works have been published in [65].

2. A novel feature-level feature fusion framework based on joint sparse representation is proposed for multi-cue visual tracking. Within this framework, unreliable features are detected and reliable features are fused on feature-level for target
representation. Based on this model, we further develop a general tracking algorithm which is able to dynamically perform feature-level fusion from various kernel spaces. These works have been published in [64, 63].

3. A novel multiple sparse representation framework with an adaptive proximity constraint, which is capable of modeling commonality and diversity of multiple features, is proposed for multi-cue visual tracking. To achieve a robust feature fusion result and accurate and informative object representation, the proposed framework simultaneously learns the commonality and diversity of the sparse patterns in multiple sparse representations which is able to exploit the consistency and complementarity of multiple visual cues for appearance modeling. A novel online multiple metric learning is also proposed to adaptively and efficiently learn the proximity constraint which is incorporated into the multiple sparse representation framework, so that the proximity constraint can further exploit the relationship between different features for adaptive close matching between the object template and the tracked target. The mistake bound for the metric learning algorithm is also derived as a part of our theoretical analysis. These works have been submitted to a journal and now is under review [66].

Both theoretical and experimental analysis show that the proposed methods can improve the tracking performance.

1.4 Overview of This Thesis

The rest of this thesis is organized as follow:

Chapter 2 presents the proposed robust joint discriminative feature learning framework for feature extraction in multi-cue visual tracking. To exploit the spatial and temporal correlation of the target samples and the background samples, the proposed feature learning framework exploits the sparse and low rank matrix decomposition technique to reveal the latent space embedded in the uncontaminated features and separate out the corrupt features encoded in the sparse matrix. To
alleviate the background distraction problem, a regularization term which measures the weighted sum of prediction loss from different visual cues is imposed. By dynamically learning the importance weights, the discriminability of different visual cues is adaptively evaluated for more reliable foreground/background discrimination. In addition, a novel feature fusion scheme is introduced to exploit the consistent and feature-specific discriminability of multiple visual cues for more robust feature fusion. The optimization procedure is also derived and the implementation details are also introduced. Experimental results on 15 challenging videos demonstrate the effectiveness of the proposed method.

Chapter 3 reports the proposed joint sparse representation-based feature-level fusion method for multi-cue visual tracking. Inspired by the multi-task joint sparse representation for visual classification, joint sparsity constraint is imposed to exploit the relationship of multiple visual cues for feature-level fusion. In addition, this chapter derives a novel unreliable feature detection scheme to detect unreliable features which do not share the same sparsity pattern in reliable features. To further exploit the non-linearity of features, this chapter also derives the kernelized framework based on kernel function for feature-level fusion in any implicit feature spaces without explicitly knowing the mapping functions. The optimization algorithm based on accelerated proximal gradient method and related complexity analysis are also given. Experimental results on both synthetic data and real videos show the effectiveness of the proposed robust feature-level fusion scheme in visual tracking.

Chapter 4 presents the proposed multiple sparse representation framework with commonality and diversity modeling for multi-cue visual tracking. To model the commonality and diversity of the multiple sparse representations of multiple visual cues, the chapter derives a sparse pattern decomposition technique which learns and fuses the shared and diverse sparse representations of multiple visual cues under a unified sparse representation framework. In addition, an adaptive proximity constraint, which is learned by a novel online multiple metric learning, is effectively incorporated into the framework, which ensures the close matching between the
tracked object and the template set with multiple visual cues. The mistake bound of the metric learning algorithm is also derived as a part of the theoretical analysis. Extensive experimental results on visual tracking benchmark and other sixteen videos show that the proposed tracker outperforms other ten state-of-the-art methods.

Chapter 5 concludes the thesis and discusses some future directions.

List of Publications

Journal Papers

1. Xiangyuan Lan, Andy J. Ma, Pong. C Yuen and Rama Chellappa, 

2. Shengping Zhang, Xiangyuan Lan, Yuankai Qi and Pong C. Yuen, 

Conference Papers


3. Renfei Liu, Xiangyuan Lan, Pong. C Yuen and Guo-Can Feng, *Robust Visual Tracking Using Dynamic Feature Weighting Based on Multiple*

List of Submitted Papers

Journal Papers

Chapter 2

Robust Joint Discriminative Feature Learning for Multi-Cue Visual Tracking

2.1 Introduction

To enhance the robustness to large appearance change, different kinds of visual cues (features) that describe different characteristics of the object, e.g. color, texture are fused and jointly exploited for appearance modeling, and a variety of feature fusion-based tracking methods have been proposed, which can be roughly divided into two categories: discriminative methods and generative methods according to their feature fusion strategies. Discriminative models combine the discriminabilities of different features to facilitate the discrimination between the target and its background. Typical approaches belonging to this category such as [33, 34, 4] are grounded on online boosting. For example, Grabner and Bischof [33] proposed an online boosting-based tracker in which a large weak classifier pool is learned and updated from various kinds of raw features, and an online boosting algorithm is employed to select and fuse weak classifiers for target/background discrimination. Along this line, more variants of boosting-based approaches such as [34, 4] are de-
developed to deal with the drifting problem. Discriminative models can alleviate the background distraction problem to some extent because different features are jointly exploited for foreground/background separation. However, the classifiers which discriminative methods exploit for feature fusion are directly learned and updated with the raw features extracted from the target samples. If the target samples are corrupted/contaminated by external variations such as occlusion, illumination, etc, the extracted features may not well reflect the intrinsic properties of the object appearance. Therefore, learning and updating classifiers using such corrupted features may degrade the fusion performance, which urges the need to remove the corrupted features or learn some uncontaminated features for robust appearance modeling.

Unlike discriminative methods, generative methods increase the representation ability of a tracker by directly fusing multiple features in object representation. To enhance the tracking robustness to large appearance variations, some strategies are adopted, e.g. using trivial templates [86, 48] to model the outliers existing in the target’s appearance, removing unreliable features [63] for robust feature-level fusion, or integrating responses from various Gabor kernels to capture the local appearance changes [153]. Since generative methods directly fuse features for model learning without mapping them to classification scores, they preserve more information of multiple features, and are more capable of accounting for appearance changes than discriminative methods. However, generative methods do not take advantage of background information for appearance modeling with multiple features, which may make them easily to be distracted by cluttered background and lead to tracking failure.

Generally speaking, discriminative and generative methods have complementary advantages in feature fusion-based appearance modeling, and the success of a visual tracker depends on both its representation ability against appearance variations and its discriminability between the target and its background. As such, the advantages of these two approaches should be exploited jointly for more robust feature fusion, so that multiple features can be employed simultaneously to describe the object ap-
pearance accurately and separate the object from background discriminatively. In
addition, different features extracted from the same object share some consistency
while each feature should also have some specific knowledge in their discriminabil-
ity. As is pointed out in [79], consistency is closely related to agreement while
feature specific knowledge provides complementarity and is related to disagree-
t. While exploring feature-specific discriminative information for feature fusion has
been shown to be effective in some methods such as online boosting [33, 34, 4], the
benefits of exploiting shared information among multiple features/modalities/views
have also been well demonstrated in some learning and classification tasks recent-
ly [137, 47, 109]. This motivates us to explore an effective strategy to jointly consider
the shared and feature-specific discriminative information for feature fusion.

Based on aforementioned motivations, we propose a novel robust joint discrimi-
native feature learning framework for object tracking using multiple visual cues.
Different from other feature fusion-based trackers which directly employ potentially
contaminated raw features and utilize the representation ability or discriminability
of different visual cues alone, the proposed method aims to learn uncontaminated
and discriminative features to jointly exploit the representation and discriminative
power of multiple visual cues for visual tracking. Within this unified framework,
feature learning is performed by simultaneously and optimally removing corrupted
features and learning reliable classifiers. As such, feature learning from multiple vi-
sual cues with corrupted feature removal offers uncontaminated features for reliable
classifier learning while discriminative classifier learning with multiple visual cues
imposes the discriminability to the learned features. Therefore, the limitations of
the generative and discriminative approaches can be compensated and the benefits
of these approaches can be combined. In addition, we incorporate a novel feature
fusion scheme into the feature learning framework to further exploit the shared and
feature-specific discriminative information for feature fusion, and the importance of
different features in target/background discrimination is also dynamically weighted
in this optimal learning framework. By jointly exploiting the learned features and
classifiers from multiple visual cues for target representation and target/background classification, the learning framework enhances the tracking performance in term of representation accuracy and discrimination reliability.

It should be noted that some hybrid approaches which attempt to combine the benefits of both the generative and discriminative approaches have been developed, e.g. [141]. Their models are developed in the context of using a single feature, while the proposed model is developed for multi-feature appearance model and can be more effectively used for features learning and fusion with multiple visual cues. Although existing single feature-based hybrid approaches may be applied to multiple features by feature concatenation, such an approach ignores different statistical properties of different features and may result in a long feature vector that may degrade the learning efficiency. The proposed method is also different from the recent developed fusion-based tracker [161] since the proposed model focuses on feature fusion while [161] focuses on tracker fusion.

The rest of the chapter is organized as follow. Section 2.2 discusses some works in pattern recognition which is related to the feature fusion scheme in the proposed feature learning model. The proposed feature learning model with the optimization procedure, and the implementation details are introduced in Section 2.3 and 2.4, respectively. Finally, the experimental evaluation and the summary are given in Section 2.5 and 2.6.

2.2 Related Work

This section discusses some related works in pattern recognition which exploit the shared and feature-specific information among multiple features/modalities/views which motivates the development of the feature fusion scheme in the proposed feature learning model.
2.2.1 Shared and Feature-Specific Information among Multiple Features/Modalities/Views for Pattern Classification

Exploiting the shared and feature-specific information among multiple features/modalities/views jointly has been shown to be beneficial for pattern classification. Yang et al. [137] proposed a multi-feature collaborative model which simultaneously models the similar and distinctive information among multiple features for image classification. Hu et al. [47] proposed to learn the shared and feature-specific structure of heterogeneous channels for RGB-D activity recognition. A multi-modal feature learning approach was proposed for learning the shared and model-specific properties for RGB-D object recognition [109]. These research findings motivate us to exploit the shared and feature-specific discriminative information for feature fusion-based visual tracking.

2.3 Proposed Model

2.3.1 Robust Joint Discriminative Feature Learning

In the $t$-th frame, let $Y_k^k = [y_{k1}^k, \ldots, y_{kn_1}^k]$ be the recently obtained target samples of the $k$-th visual cues, and $n_1$ denote the number of target samples in the sample set. Since large appearance variations, e.g. occlusion, illumination may occur during tracking, the captured samples may be contaminated/corrupted. To ensure the robustness of the learned features, explicitly separating out the corrupted samples is essential. Therefore, one objective of the learning model is to learn the uncontaminated features while separating out the corrupted features as follows:

$$Y_k^k = X_k^k + E_k^k, \quad k = 1, \ldots, K$$

(2.3.1)

where $X_k^k$ and $E_k^k$ are the learned uncontaminated features and the separated corrupted features, respectively, and $K$ is the number of visual cues. The background samples near the target position, also known as local context information in the
current frame, may share some similarity with the recently obtained target samples, such as illumination conditions. Exploiting such context information has been shown to be beneficial for tracking [35]. Besides, the target samples in recent frames are temporally correlated, and mining the latent structure embedded in the target samples can facilitate revealing the intrinsic characteristic of the target’s features [93]. To further exploit the spatial and temporal correlations of the background and the target samples for feature learning while separating out the corrupted features from multiple visual cues, we cast the objective discussed above into the following rank and sparsity minimization problem with the sample set of multiple visual cues:

$$\min_{\{X^k, E^k\}_{k=1}^K} \sum_{k=1}^K \{\text{rank}(X^k) + \lambda_1 \|E^k\|_1\}$$

s.t. \(Y^k = X^k + E^k, \ k = 1, \ldots, K\) \hfil (2.3.2)

where \(Y^k_B = [y^k_{m+1}, \ldots, y^k_N]\) is the nearby background samples in current frame, \(N\) is the total number of samples. \(Y^k = [Y^k_F, Y^k_B], X^k = [X^k_F, X^k_B]\) and \(E^k = [E^k_F, E^k_B]\) is the original feature set, the learned feature set and the separated feature set of the target samples and the background samples in the \(k\)-th visual cue, respectively. With the same merit of RPCA [12], the rank minimization term is able to uncover the shared latent space embedded in the samples of different visual cues which characterizes intrinsic properties of uncontaminated features of the target and the background, while the sparsity regularization is employed to model the outliers that exist in the corrupted features.

Although the feature learning scheme in (2.3.2) is able to learn uncontaminated informative features from multiple visual cues for target representation via joint low-rank and sparse matrix decomposition, it cannot guarantee that the target and background samples can be well discriminated with the learned features. As such, appearance modeling with the learned features may suffer the loss of discriminability, which may lead to the background distraction problem. To strengthen the discriminability of the learned features while modeling such different discriminabilities of different visual cues for robust feature fusion, we impose the discriminability regularization which measures the prediction loss using the learned classifiers to the
feature learning process as follows:

\[
\min_{\{w^k, b^k, \beta^k\}_{k=1}^K} \sum_{k=1}^K ((\beta^k)^2 \| (X^k)^T w^k + b^k - L_k^k \|^2 + \lambda_2 \| w^k \|^2)
\]

s.t. \( \sum_{k=1}^K \beta^k = 1, \ \beta^k \geq 0, \ k = 1, \ldots, K \) \hspace{1cm} (2.3.3)

where \( L_k = [L_{k1}^1, \ldots, L_{kN}^k]^T \) is the label vector, \( L_{ki}^k = +1(-1) \) means that the \( i \)-th sample of the \( k \)-the visual cue \( x_{ki}^k \) belongs to the target (background) class, \( \beta^k \) is the importance weight of the prediction loss corresponding to the \( k \)-th visual cue, \( w^k \in \mathbb{R}^{d_k} \), \( b^k \in \mathbb{R} \), whose elements are all 1s, \( b^k \in \mathbb{R} \), and \( d_k \) is the dimension of the \( k \)-th visual cue. From (2.3.3), we can see that the discriminability regularization aims to minimize the weighted sum of the prediction loss of the learned features in different visual cues based on multiple linear classifiers \( \{w^k, b^k\}_{k=1}^K \) which are learned jointly.

Therefore, it ensures the learned features in multiple visual cues for the target and the background samples can be linearly separated as well as possible, which is able to enhance the discriminability of the tracking model and hence alleviates the background distraction problem. Moreover, dynamically learning and updating the importance weights during tracking allows the discriminative powers of different visual cues to be adaptively evaluated, which guarantees that more discriminative features play more important roles in target/background discrimination. Here we use \((\beta^k)^2\) instead of \(\beta^k\) for feature fusion because we want to ensure all the weights are positive which avoids the trivial solution that the weight corresponding to the lowest prediction loss is 1, and 0 otherwise.

To further exploit the shared and feature-specific discriminative information of multiple visual cues, we introduce the following objective function into the proposed feature learning framework:

\[
\min_{\{w^k, b^k\}_{k=1}^K, L^*} \sum_{k=1}^K \theta^k \| (X^k)^T w^k + b^k - L_k^* \|^2
\]

where the \( L^* = [L_1^*, \ldots, L_N^*]^T \), and \( L_{i}^* \) is the learned consensus classification score of different visual cues in the \( i \)-th training sample, which reflects the consistent discriminative information from different visual cues. Different from other discriminative feature fusion models which enforce different visual cues to share the same
classification score [140] or to be with diverse discriminative information [81], the objective function softly regularizes the classification scores towards the consensus while enabling them to have some disagreement with the consensus. Therefore, both the consistent and feature-specific information among multiple visual cues are explicitly and jointly employed for learning informative features and reliable classifiers. Moreover, the disagreement with the consensus can be controlled by $\theta^k$, and larger (less) $\theta^k$ will promote less (larger) disagreement.

**Unifying them all together.** Based on the aforementioned analysis, we formulate the objectives as mentioned above into an unified joint discriminative feature learning framework in which uncontaminated and corrupted features, classifier parameters of multiple visual cues, denoted as $\Omega = \{L^*, X^k, E^k, w^k, b^k, \beta^k | k = 1, \ldots, K\}$ are jointly estimated as follows:

\[
\min_{\Omega} \sum_{k=1}^{K} \{\|X^k\|_* + \lambda_1\|E^k\|_1 + \frac{\lambda_2}{2}\|w^k\|^2_2 \\
+ \frac{\alpha_1 (\beta^k)^2}{2N} \|(X^k)^T w^k + 1b^k - L^k\|^2_2 \\
+ \frac{\alpha_2 \theta^k}{2N} \|(X^k)^T w^k + 1b^k - L^*\|^2_2\} \\
\text{s.t.} \quad Y^k = X^k + E^k, \quad \sum_{k=1}^{K} \beta^k = 1 \\
\beta^k \geq 0, \quad k = 1, \ldots, K
\] (2.3.5)

where $\alpha_1$ and $\alpha_2$ are the nonnegative parameters associating with different objective functions, and the constant $\frac{1}{2}$ is used for simplifying deductions. Since the rank minimization in (2.3.2) is a NP-hard problem, we employ the standard approach [12] and relax this problem by using nuclear norm $\| \cdot \|_*$ instead. The optimization procedure for (2.3.5) is derived in Section 2.3.2.

**2.3.2 Optimization**

The objective function in (2.3.5) is convex with respect to one of these four blocks $\{X^k, E^k\}_{k=1}^{K}$, $\{w^k, b^k\}_{k=1}^{K}$, $L^*$ and $\{\beta^k\}_{k=1}^{K}$ when the other three blocks are fixed, and it’s difficult to derive the analytical solution to (2.3.5). Therefore, we derive an iterative optimization algorithm to solve the problem. To make the problem separ-
Algorithm 1: Optimization Algorithm for (2.3.5)

**Input:** Sample matrix \( \{Y_k\}_{k=1}^{K} \), label vector \( \{L_k\}_{k=1}^{K} \), sample number \( N \) and feature number \( K \)

**Output:** \( \{X^{k,i}, E^{k,i}, w^{k,i}, b^{k,i}, \beta^{k,i}\}_{k=1}^{K}, L^* \)

**Initialization:** \( i \leftarrow 1, X^{k,i} \leftarrow Y^k, E^{k,i} \leftarrow 0, \beta^{k,i} \leftarrow \frac{1}{K}, w^{k,i} \leftarrow 0, b^{k,i} \leftarrow 0, k = 1, ...K, a_t \leftarrow \frac{\alpha_t}{N}, t = 1 \) or \( 2 \)

while stopping conditions are not satisfied do

Update \( \{X^{k,i+1}, E^{k,i+1}\}_{k=1}^{K} \) via Algorithm (2)

Update \( \{w^{k,i+1}, b^{k,i+1}\}_{k=1}^{K} \) via solving (2.3.10)

Update \( L^*, i+1 \) via solving (2.3.11)

Update \( \{\beta^{k,i+1}\}_{k=1}^{K} \) via solving (2.3.12)

\( i \leftarrow i + 1 \)

Check stopping conditions

end

ble, we introduce the auxiliary variables \( \{Z^k\}_{k=1}^{K} \) to replace \( \{X^k\}_{k=1}^{K} \) in the nuclear norm \( \|\cdot\|_* \) of (2.3.5). Accordingly, \( \{\forall k, X^k = Z^k\} \) act as additional constraints. Let \( C \) be the constraint set of (2.3.5) on \( \{\beta^k\}_{k=1}^{K} \), and \( a_i = \frac{\alpha_i}{N} \) for \( i = 1 \) or \( 2 \). Then the augmented Lagrange function of (2.3.5) \( \mathcal{L}_{\Omega \in C} \) is

\[
\sum_{k=1}^{K} \left( \|Z^k\|_* + \lambda_1\|E^k\|_1 + \Phi(\Lambda^k, Y^k - X^k - E^k) + \frac{a_1(\beta^k)^2}{2}\|X^k\|_F^2 + \|b^k - L^\ast\|_2^2 + \Phi(\Gamma^k, X^k - Z^k) + \frac{\alpha_2(\theta^k)^2}{2}\|X^k\|_F^2 + \|b^k - L^\ast\|_2^2 + \lambda_2\|w^k\|_2^2 \right) \quad (2.3.6)
\]

with the definition \( \Phi(A, B) = \langle A, B \rangle + \mu\|B\|_F^2 \), where \( \mu \) is a positive penalty scalar, \( \langle A, B \rangle = trace(A^TB) \) and, \( \{\Lambda^k, \Gamma^k\}_{k=1}^{K} \) are the Lagrangian multipliers.

Based on (2.3.6), the solutions to (2.3.5) can be obtained by iteratively solving the subproblems of (2.3.5) in which an inner loop procedure is employed to solve \( \{X^k, E^k\} \).

\( \{X^k, E^k\}\)-subproblem: Keeping other variables fixed, we obtain \( X^k, E^k, k = 1, \ldots, K \) using Alternating Direction Method of Multipliers (ADMM) [11]. In the \( (j + 1) \)-th
step of ADMM, by some algebraic manipulations, $E^{k,j+1}$ and $Z^{k,j+1}$ are obtained as

$$Z^{k,j+1} = \arg\min_{Z^k} \frac{1}{2} \|Z^k - Q^{k,j}\|_F^2 + \frac{1}{\mu} \|Z^k\|_* = J_\mu(Q^{k,j})$$

$$E^{k,j+1} = \arg\min_{E^k} \frac{1}{2} \|E^k - P^{k,j}\|_F^2 + \frac{1}{\lambda_1} \|E^k\|_1 = S_{\lambda_1}(P^{k,j})$$

(2.3.7)

where $Q^{k,j} = X^{k,j} + \frac{\Gamma^{k,j}}{\mu}$, and $P^{k,j} = Y^{k,j} - X^{k,j} + \frac{\Lambda^{k,j}}{\mu}$. $S_\mu(\cdot)$ is a soft-thresholding operator such that $S_\mu(A)_{m,n} = \text{sign}(A_{m,n}) \cdot \max(|A_{m,n}| - \mu, 0)$, and $J_\mu(\cdot)$ is the singular value soft-thresholding operator such that $J_\mu(A) = U_A S_\mu(\Sigma_A) V_A^T$ where $U_A \Sigma_A V_A^T$ is the singular value decomposition of $A$. Then by taking partial derivatives of (2.3.6) with respect to $X^k$, we obtain

$$X^{k,j+1} = [G_1^{k,j}]^{-1} G_2^{k,j}$$

(2.3.8)

where $G_1^{k,j} = (a_1(\beta^k)^2 + a_2)w^k(w^k)^T + 2\mu I$, and $G_2^{k,j} = a_1(\beta^k)^2w^k((L^k)^T - b^k 1^T) + a_2\theta^k w^k((L^*)^T - b^k 1^T) + \mu(Y^{k,j+1} - E^{k,j+1} - Z^{k,j+1}) + \Lambda^{k,j} - \Gamma^{k,j}$.

After $\{X^{k,j+1}, Z^{k,j+1}, E^{k,j+1}\}_{k=1}^K$ are obtained, the Lagrangian multipliers are updated as follows:

$$\Lambda^{k,j+1} = \Lambda^{k,j} + \mu(Y^{k,j+1} - X^{k,j+1} - E^{k,j+1})$$

$$\Gamma^{k,j+1} = \Gamma^{k,j} + \mu(X^{k,j+1} - Z^{k,j+1})$$

(2.3.9)

The ADMM algorithm iteratively updates the optimal variables and the Lagrangian multipliers in the inner loop of the whole optimization algorithm until $\|Y^k - X^k - E^k\|_F^2 < \epsilon\|Y^k\|_F^2$ or the maximum iteration number is reached. Then with $\{X^{k,i}, E^{k,i}\}_{k=1}^K$ in the $i$-th step of the outer loop, we can solve the subproblem with respect to other variables.

$\{w^k, b^k\}$-subproblem: With other variables fixed, it is equivalent to solve the following problem:

$$\min_{w^k, b^k} \frac{a_1(\beta^k)^2}{2} \|(X^k)^T w^k + 1b^k - L^k\|_2^2$$

$$+ \frac{a_2\theta^k}{2} \|(X^k)^T w^k + 1b^k - L^*\|_2^2 + \frac{\lambda_2}{2} \|w^k\|_2^2$$

(2.3.10)

which is an unconstrained quadratic programming problem and can be solved by some standard optimization techniques, e.g. conjugate gradient descent method.
**Algorithm 2: Solver for \( \{X^k, E^k\}\)-subproblem**

**Input**: Sample matrix of \( k \)-th visual cue \( Y^k \), label vector \( L^k \), sample number \( N \), other optimal variable with fixed values \( w^k, b^k, \beta^k, L^* \), initial values \( X_{k,i}^{k,i-1}, E_{k,i}^{k,i-1} \), from \((i-1)\)-th iteration in Algorithm 1

**Output**: \( X_{k,i}^{k,i}, E_{k,i}^{k,i} \)

**Initialization**:

\[
j \leftarrow 1, X_{k,j}^{k,j} \leftarrow X_{k,i}^{k,i-1}, E_{k,j}^{k,j} \leftarrow E_{k,i}^{k,i-1}, Z_{k,j}^{k,j} \leftarrow X_{k,i}^{k,i-1}, \Gamma_{k,j}^{k,j} \leftarrow 0, \Lambda_{k,j}^{k,j} \leftarrow 0
\]

**while stopping conditions are not satisfied** do

- Update \( Z_{k,j+1}^{k,j+1} \) and \( E_{k,j+1}^{k,j+1} \) via (2.3.7)
- Update \( X_{k,j+1}^{k,j+1} \) via (2.3.8)
- Update \( \Lambda_{k,j+1}^{k,j+1} \) and \( \Gamma_{k,j+1}^{k,j+1} \) via (2.3.9)
- \( j \leftarrow j + 1 \)
- Check stopping conditions

**end**

\( X_{k,i}^{k,i} \leftarrow X_{k,j}^{k,j}, E_{k,i}^{k,i} \leftarrow E_{k,j}^{k,j} \)

**L* -subproblem**: Similar to the \( \{w^k, b^k\}\)-subproblem, some standard optimization techniques can be used to solve the following unconstrained quadratic programming problem:

\[
\min_{L^*} \sum_{k=1}^{K} \frac{\alpha w^k}{2} \| (X^k)^T w^k + b^k - L^* \|^2_2
\]  

(2.3.11)

**\( \beta^k \)-subproblem**: Let \( R^k = \frac{a^k}{2} \| (X^k)^T w^k + b^k - L^k \|^2_2 \). Then this subproblem can be rewritten as

\[
\min_{\beta^k} \sum_{k=1}^{K} (\beta^k)^2 R^k \\
\text{s.t.} \sum_{k=1}^{K} \beta^k = 1, \quad \beta^k \geq 0, \quad k = 1, \ldots, K
\]  

(2.3.12)

which is a quadratic programming problem with linear constraints. Based on its Lagrange function, the solution can be derived as \( \beta^{k'} = \frac{(R^{k'})^{-1}}{\sum_{k=1}^{K} (R^k)^{-1}} \). We iteratively solve these four subproblems until the relative change of the valuables in the adjacent iterations are less than a predefined threshold or the maximum iteration...
tion number is reached. The optimization algorithm for (2.3.5) and the solver for \( \{X^k, E^k\}\)-subproblem are summarized in Algorithm 1 and 2, respectively.

2.4 Implementation Details

2.4.1 Target Representation

For the sake of robustness and effectiveness [155], we adopt the sparse representation scheme [87] for target representation. To further enhance the adaptivity of the proposed tracker, we augment the learned features for each visual cue \( A^k \) with the recent obtained important target samples \( R^k_F \) to construct a template set \( D^k = [X^k_F, R^k_F] \) where \( R^k_F \) is updated adaptively similar to [87]. Then we can obtain the sparse representations of the target candidates of each visual cue \( u^k_i, i = 1, \ldots, m, k = 1, \ldots, K \) as follows:

\[
 u^k_i = \arg \min_u \left\| t^k_i - D^k u \right\|_2^2 + \lambda \| u \|_1
\]  

(2.4.13)

where \( \lambda \) is the tradeoff between the reconstruction error and the sparseness, \( t^k_i \) is the \( k \)-th visual cue of the \( i \)-th target candidate sampled by particle filter in every frame.

2.4.2 Observation Likelihood for Particle Filtering

The proposed tracking algorithm is developed in the particle filtering framework. After obtaining the optimal solution in (2.4.13), we derive the observation likelihood based on the learned features and classifiers as follows:

\[
p(o_t|s_t) \propto \exp(-\sum_{k=1}^{K} \| t^k - D^k u^k \|_2^2 \quad \text{\( -\rho \sum_{k=1}^{K} \beta^k((w^k)^T D^k u^k + b^k) - 1) \)}}
\]  

(2.4.14)

Here we use the joint decisions based on the reconstruction error and classification reliability of multiple visual cues to find out the true state of the target. This is because a good candidate should have high confidence to be the label of the target (+1) while it should also have low reconstruction error with multiple features.
Therefore, both the representation abilities and discriminabilities of multiple visual cues are jointly exploited for target state estimation. It should be noted that the classification reliability is based on the reconstructed samples from the learned features instead of using the original samples. This is because the classifiers are estimated using the learned features, and it is more reliable to perform classification in the learned feature space.

2.5 Experiments

In this section, we report the experimental results of the proposed tracker quantitatively and qualitatively.

2.5.1 Experimental Setting

In this section, we evaluate the proposed tracker using fifteen sequences which covers different kinds of challenging factors including cluttered background, illumination variations, partial occlusion, pose variation, etc. We compared the proposed tracking algorithm with other ten state-of-the-art trackers which include discriminative multiple feature trackers: OAB [33], SemiT [34], MIL [4], generative trackers which explicitly model the noise/outliers: L1T [87], MTT [158], feature learning-based trackers: IVT [93], and other state-of-the-art methods: CT [149], DFT [95], LOT [90], STRUCK [37]. We use the source codes provided by the authors of these papers and set them to be with the same initialization parameters for fair comparison.

We empirically set $\alpha_1$, $\alpha_2$, $\lambda_1$, and $\lambda_2$ to be 0.25, 0.25, 0.1, 0.01, respectively. These parameters are selected based on the qualitative result of cross-validation in small number of videos. All $\theta_k(k = 1, \ldots, K)$ are set to be 1. The training samples $Y^k(k = 1, \ldots, K)$ consist of the tracking results of the initial 5 and recent 10 frames, and 10 background samples in the current frame. We implement the $\ell_1$ tracker [87] with multiple features to track the target in the initial 15 frames to get target
samples for feature learning. We use HOG [20] as global features and covariance descriptors [108] with log-Euclidean metric [2] as local features which describe the 2-by-2 non-overlapping parts. So totally 5 kinds of raw visual features are used. We use the HOG feature extractors in the VLFeat toolbox ¹ to extract the HOG features. We set the cell size to be 8 and the number of gradient orientation to be 9. We follow the setting in [49] to extract the covariance descriptors from 4 blocks where the size of each block is 8-by-8.

2.5.2 Experimental Results

We adopt two metrics: center location error and success rate for quantitative comparison. The center location error is the Euclidean distance between the center of bounding box and the ground truth. The VOC overlapping rate is defined as

\[
\frac{\text{area}(\text{ROI}_T \cap \text{ROI}_G)}{\text{area}(\text{ROI}_T \cup \text{ROI}_G)},
\]

where \(\text{ROI}_T\) and \(\text{ROI}_G\) are the bounding boxes of the tracker and ground-truth. The tracking result of each frame is considered as a success if the overlapping rate is larger than 0.5.

Tables 2.1 and 2.2 record the center location error and the success rate on the 15 videos, respectively. The results show that the proposed tracker outperforms other compared trackers on most videos in terms of both two evaluation metrics such that the center location error of the proposed method rank in top three on 12 videos while the success rate ranks in top three on 14 videos. Particularly, the proposed method demonstrates its superior performance in videos which cover cluttered background (e.g. Trellis, Crossing, Bolt), occlusion (e.g. DavidOutdoor, Faceocc), and illumination (e.g. Shaking, Skating1). This is because the feature learning is performed by simultaneously removing contaminated features and imposing discriminability which enables the appearance model to be less sensitive to target samples contaminated by occlusion and large illumination variation and more discriminative under cluttered background. Figure 2.1 shows the quantitative frame-by-frame comparison result on some challenging videos, i.e. Trellis, Skating1, DavidOutdoor and Bolt. We can

¹http://www.vlfeat.org/
Table 2.1: Center Location Error. The best three results are shown in red, green and blue.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>STRUCK</th>
<th>LOT</th>
<th>OAB</th>
<th>SemiT</th>
<th>L1T</th>
<th>MTT</th>
<th>MIL</th>
<th>IVT</th>
<th>DFT</th>
<th>CT</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossing</td>
<td>3</td>
<td>36.6</td>
<td>4.6</td>
<td>3.8</td>
<td>62.9</td>
<td>56.5</td>
<td>2.9</td>
<td>2</td>
<td>21.8</td>
<td>3.2</td>
<td>3</td>
</tr>
<tr>
<td>Car11</td>
<td>1.1</td>
<td>31.4</td>
<td>2.9</td>
<td>2.5</td>
<td>1.4</td>
<td>1.8</td>
<td>44</td>
<td>8.9</td>
<td>59</td>
<td>119.3</td>
<td>2.7</td>
</tr>
<tr>
<td>Car4</td>
<td>8.8</td>
<td>167.4</td>
<td>95.9</td>
<td>152.7</td>
<td>77</td>
<td>22.6</td>
<td>50.2</td>
<td>2.4</td>
<td>62.5</td>
<td>86</td>
<td>9.2</td>
</tr>
<tr>
<td>Animal</td>
<td>5.3</td>
<td>64.9</td>
<td>6.8</td>
<td>61.6</td>
<td>24.1</td>
<td>19.1</td>
<td>100.2</td>
<td>182.6</td>
<td>98.3</td>
<td>246.2</td>
<td>7</td>
</tr>
<tr>
<td>DavidIndoor</td>
<td>42.3</td>
<td>24.4</td>
<td>22.2</td>
<td>12.2</td>
<td>46.8</td>
<td>13.8</td>
<td>32.5</td>
<td>1.4</td>
<td>1.8</td>
<td>44.3</td>
<td>42.2</td>
</tr>
<tr>
<td>Crossing</td>
<td>83.2</td>
<td>110.4</td>
<td>43.2</td>
<td>288.3</td>
<td>159.2</td>
<td>293.8</td>
<td>139.6</td>
<td>247.6</td>
<td>174.6</td>
<td>150.9</td>
<td>10.2</td>
</tr>
<tr>
<td>Subway</td>
<td>4.1</td>
<td>5</td>
<td>113.2</td>
<td>105.6</td>
<td>148.1</td>
<td>166.3</td>
<td>8</td>
<td>131.3</td>
<td>3.2</td>
<td>11.6</td>
<td>4.9</td>
</tr>
<tr>
<td>Faceocc</td>
<td>19.3</td>
<td>35.2</td>
<td>25.1</td>
<td>75.1</td>
<td>17.8</td>
<td>21.6</td>
<td>30.1</td>
<td>18.9</td>
<td>23.8</td>
<td>26.3</td>
<td>18.4</td>
</tr>
<tr>
<td>Faceocc2</td>
<td>6</td>
<td>15</td>
<td>19.8</td>
<td>50.9</td>
<td>12.9</td>
<td>10.2</td>
<td>13.9</td>
<td>7.8</td>
<td>8.1</td>
<td>18.9</td>
<td>9.3</td>
</tr>
<tr>
<td>Walking2</td>
<td>10.7</td>
<td>64.5</td>
<td>28.7</td>
<td>11.9</td>
<td>4.5</td>
<td>3.5</td>
<td>60.2</td>
<td>2</td>
<td>28.5</td>
<td>58.3</td>
<td>7.3</td>
</tr>
<tr>
<td>Bolt</td>
<td>399.3</td>
<td>12.1</td>
<td>256.5</td>
<td>363.7</td>
<td>409</td>
<td>409.2</td>
<td>394.1</td>
<td>397.5</td>
<td>367.6</td>
<td>364.2</td>
<td>8.9</td>
</tr>
<tr>
<td>Shaking</td>
<td>30.1</td>
<td>82.7</td>
<td>191.6</td>
<td>275</td>
<td>109.7</td>
<td>97.3</td>
<td>24</td>
<td>86.1</td>
<td>26.3</td>
<td>79.4</td>
<td>9.7</td>
</tr>
<tr>
<td>DavidOutdoor</td>
<td>106.7</td>
<td>9.7</td>
<td>83.5</td>
<td>237</td>
<td>89.8</td>
<td>341.8</td>
<td>30</td>
<td>51.9</td>
<td>51</td>
<td>88.6</td>
<td>7.2</td>
</tr>
</tbody>
</table>

Table 2.2: Success Rate. The best three results are shown in red, green and blue.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>STRUCK</th>
<th>LOT</th>
<th>OAB</th>
<th>SemiT</th>
<th>L1T</th>
<th>MTT</th>
<th>MIL</th>
<th>IVT</th>
<th>DFT</th>
<th>CT</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossing</td>
<td>0.96</td>
<td>0.33</td>
<td>0.84</td>
<td>0.88</td>
<td>0.25</td>
<td>0.23</td>
<td>0.99</td>
<td>0.24</td>
<td>0.66</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Car11</td>
<td>1</td>
<td>0.56</td>
<td>0.95</td>
<td>0.93</td>
<td>1</td>
<td>1</td>
<td>0.18</td>
<td>0.7</td>
<td>0.34</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Car4</td>
<td>0.41</td>
<td>0.05</td>
<td>0.28</td>
<td>0.25</td>
<td>0.3</td>
<td>0.32</td>
<td>0.28</td>
<td>1</td>
<td>0.26</td>
<td>0.28</td>
<td>0.45</td>
</tr>
<tr>
<td>Animal</td>
<td>1</td>
<td>0.06</td>
<td>0.96</td>
<td>0.86</td>
<td>0.72</td>
<td>0.87</td>
<td>0.13</td>
<td>0.03</td>
<td>0.31</td>
<td>0.04</td>
<td>0.99</td>
</tr>
<tr>
<td>DavidIndoor</td>
<td>0.24</td>
<td>0.16</td>
<td>0.15</td>
<td>0.21</td>
<td>0.7</td>
<td>0.29</td>
<td>0.25</td>
<td>0.8</td>
<td>0.24</td>
<td>0.44</td>
<td>0.59</td>
</tr>
<tr>
<td>Trellis</td>
<td>0.78</td>
<td>0.32</td>
<td>0.18</td>
<td>0.2</td>
<td>0.16</td>
<td>0.2</td>
<td>0.24</td>
<td>0.32</td>
<td>0.52</td>
<td>0.36</td>
<td>0.81</td>
</tr>
<tr>
<td>Skating1</td>
<td>0.37</td>
<td>0.28</td>
<td>0.34</td>
<td>0.08</td>
<td>0.13</td>
<td>0.14</td>
<td>0.1</td>
<td>0.08</td>
<td>0.16</td>
<td>0.11</td>
<td>0.28</td>
</tr>
<tr>
<td>Mountain-bike</td>
<td>0.86</td>
<td>0.7</td>
<td>0.92</td>
<td>0.29</td>
<td>0.93</td>
<td>0.97</td>
<td>0.58</td>
<td>0.99</td>
<td>0.35</td>
<td>0.17</td>
<td>0.97</td>
</tr>
<tr>
<td>Subway</td>
<td>0.94</td>
<td>0.7</td>
<td>0.22</td>
<td>0.38</td>
<td>0.23</td>
<td>0.08</td>
<td>0.81</td>
<td>0.21</td>
<td>0.99</td>
<td>0.8</td>
<td>0.87</td>
</tr>
<tr>
<td>Faceocc</td>
<td>1</td>
<td>0.3</td>
<td>0.91</td>
<td>0.71</td>
<td>1</td>
<td>1</td>
<td>0.77</td>
<td>0.98</td>
<td>0.81</td>
<td>0.86</td>
<td>1</td>
</tr>
<tr>
<td>Faceocc2</td>
<td>1</td>
<td>0.35</td>
<td>0.76</td>
<td>0.56</td>
<td>0.81</td>
<td>0.9</td>
<td>0.94</td>
<td>0.92</td>
<td>1</td>
<td>0.76</td>
<td>0.99</td>
</tr>
<tr>
<td>Walking2</td>
<td>0.46</td>
<td>0.39</td>
<td>0.4</td>
<td>0.38</td>
<td>0.98</td>
<td>0.99</td>
<td>0.39</td>
<td>1</td>
<td>0.39</td>
<td>0.39</td>
<td>0.44</td>
</tr>
<tr>
<td>Bolt</td>
<td>0.02</td>
<td>0.55</td>
<td>0.04</td>
<td>0.07</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>Shaking</td>
<td>0.17</td>
<td>0.08</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
<td>0.23</td>
<td>0.01</td>
<td>0.82</td>
<td>0.04</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>DavidOutdoor</td>
<td>0.34</td>
<td>0.95</td>
<td>0.34</td>
<td>0.18</td>
<td>0.46</td>
<td>0.1</td>
<td>0.69</td>
<td>0.64</td>
<td>0.75</td>
<td>0.35</td>
<td>0.93</td>
</tr>
</tbody>
</table>

see that compared with most of other trackers, the proposed tracker can maintain a relatively low tracking error throughout these videos. We can also see that although the proposed tracker is able to track the target throughout the videos FaceOcc2 and Walking2 which encounter in-plane rotation and large scale change, respectively,
Figure 2.1: Quantitative frame-by-frame comparison of 11 trackers on 4 Challenging videos in terms of center location error

does not achieve good quantitative results. This is mainly because the rotation and scale state of the tracked target is not well modeled by the particle filter used in the proposed tracker. Since we adopt the sparse representation scheme for target representation and the iterative algorithm for optimization, the proposed tracker can not run in real-time speed. The running time is about 3 frames per second.

Figure 2.2 illustrates some qualitative results on some typical frames including cluttered background and appearance variations caused by illumination, pose, and occlusion. By properly fusing the learned features from multiple visual cues, the proposed tracker is more stable under some large illumination variations (e.g.
Figure 2.2: Qualitative results on some typical frames including some challenging factors (video name). (a) Illumination (Car4, Car11, Trellis). (b) Pose (Shaking, Mountain-bike, Skating1). (c) Occlusion (Subway, DavidOutdoor, Faceocc2). (d) Cluttered background (Crossing, Animal, Bolt)

Car4 #196, Trellis #242). And explicitly modeling the discriminability of the learned features facilitates the resistance to cluttered background (e.g. Shaking #60, Mountain-bike #172). Removing contaminated/corrupted feature for appearance modeling enhances the robustness to occlusion (e.g. DavidOutdoor #88, Faceocc2 #162).

2.6 Conclusion

In this chapter, we propose a novel feature learning framework called robust joint discriminative feature learning for visual tracking with multiple features. By removing the corrupted/contaminated features, introducing the discriminabilities and explicitly modeling the consistency and complementarity in the discriminabilities of multiple visual cues for feature learning in an optimal unified framework, the proposed framework is able to jointly exploit the representation abilities and discrim-
inabilities from multiple visual cues for appearance modeling with multiple features. Extensive comparison experiments with other ten state-of-the-art trackers show its effectiveness and superior performance.
Chapter 3

Joint Sparse Representation and Robust Feature-Level Fusion for Multi-Cue Visual Tracking

3.1 Introduction

Feature fusion can be performed at different levels, and according to the fusion level, existing multi-cue tracking algorithms can be roughly divided into two categories: score-level and feature-level. Score-level approaches combine classification score corresponding to different visual features to perform foreground and background classification. Methods such as online boosting [33] [34], multiple kernel boosting [135] and online multiple instance boosting [4] belong to this category. However, the Data Processing Inequality (DPI) [19] indicates that the feature-level contains more information than the classifier level. Therefore, feature-level fusion should be performed to take advantage of more informative cues for tracking. A typical feature-level fusion method is to concatenate different feature vectors to form a single vector [127]. But such a method may result in a high-dimensional feature vector which may degrade tracking efficiency. In addition, ignoring the incompatibility of heterogeneous features may cause performance degradation. Another feature-
level fusion method is multiple kernel learning (MKL) [132], which aims to learn a weighted combination of different feature kernels according to their discriminative power. Since some features extracted from a target may be corrupted due to some unexpected variations, performance of the pre-learned classifier based on MKL may be affected by such features. Moreover, because not all cues/features are reliable, combining all the features may not improve the tracking performance. As such, the dynamic selection/combination of visual cues/features is required for multi-cue tracking.

Recently, multi-task joint sparse representation (MTJSR) [142] has been proposed for feature-level fusion for object classification and promising results have been reported. In MTJSR, the class-level joint sparsity patterns among multiple features are discovered by using a joint sparsity-inducing norm. Therefore, the relationship among different visual cues can be discovered by the joint sparsity constraint. Moreover, high-dimensional features are represented by low-dimensional reconstruction weights for efficient fusion. However, MTJSR is derived based on the assumption that all representation tasks are closely related and share the same sparsity pattern, which may not be valid in tracking applications, since unreliable features may exist due to appearance and other variations of a tracked target. For example, when a tracked object is partially occluded, its local features extracted from the occluded part(s) may not be relevant to or well represented by the corresponding feature dictionary. In such cases, reconstruction coefficients of such features may not demonstrate the sparsity pattern, which is an intrinsic characteristic of sparse representation as indicated in [126] and [87]. In addition, since unreliable features are not relevant to its corresponding feature dictionary, the reconstruction error may be large. Fusing such features with large reconstruction error may not be able to describe the object appearance well, which may degrade the tracking performance. In [142], a weighted scheme based on LPBoost [29] is adopted to measure the confidence of different features on a validation set for fusion. But in visual tracking, the variation of the tracked target cannot be predicted, and it is not reasonable to
measure the confidence of different features based on off line validation set.

In order to address the above mentioned issues in feature-level fusion based trackers, we propose a new tracking algorithm called Robust Joint Sparse Representation-based Feature-level Fusion Tacker (RJSRFFT). We integrate feature selection, fusion and representation into a unified optimization model. Within this model, unreliable features are detected and feature-level fusion is simultaneously performed on selected reliable features. Therefore, the negative effect from unreliable features can be removed, and the relationship of reliable features can be exploited to select representative templates for a sparse representation of the tracked target, which processes the advantage of sparse trackers. Based on the proposed model, we also develop a general tracking algorithm called Robust Joint Kernel Sparse Representation based Feature-level Fusion Tacker (K-RJSRFFT) to dynamically perform fusion of different features from different kernel spaces, which is able to utilize the non-linearity in features to enhance the tracking performance.

The rest of this chapter is organized as follows. In Section 3.2, we review some joint sparse representation-based methods for computer vision which is related to the proposed method. In Section 3.3, we will present the proposed feature-level fusion tracking algorithm. In Section 3.4, we discuss some implementation details of the proposed algorithm. Experiment results and summary are presented in Sections 3.5 and 3.6, respectively.

### 3.2 Related Work

In this section, we review some joint sparse representation methods related to our proposed tracking algorithm.

#### 3.2.1 Multi-Task Joint Sparse Representation

Multi-task learning aims to improve the overall performance of related tasks by exploiting the cross-task relationships. Yuan et al. [142] formulated linear repre-

3.3 Robust Feature-Level Fusion for Multi-Cue Tracking

This section presents the details of the proposed feature-level fusion tracking algorithm. The proposed method is developed based on the particle filter framework, and consists of two major components: feature-level fusion based on joint sparse representation and detecting unreliable visual cues for robust fusion.

3.3.1 Particle Filter

Our tracking algorithm is developed under the framework of sequential Bayesian inference. Let \( l_t \) and \( z_t \) denote the latent state and observation at time \( t \), respectively. Given a set of observation \( Z_t = \{z_t, t = 1, ..., T\} \) up to Frame \( T \), the true posterior \( P(l_t|Z_t) \) is approximated by a particle filter with a set of particles \( l^i_t, i = 1, ..., n \). The latent state variable \( l_t \), which describes the motion state of target, is estimated using:

\[
\hat{l}_t = \arg \max_{l^i_t} P(l^i_t|Z_t) \quad (3.3.1)
\]

The tracking problem is thus formulated as recursively estimating of the posterior probability \( P(l_t|Z_t) \),

\[
P(l_t|Z_t) \propto P(z_t|l_t) \int P(l_t|l_{t-1})P(l_{t-1}|Z_{t-1})dl_{t-1} \quad (3.3.2)
\]
where \( P(l_t | l_{t-1}) \) denotes the motion model, and \( P(z_t | l_t) \) the observation model. We use the same motion model as in [168], and define the observation model using the proposed tracking algorithm, which will be described in the following subsection.

### 3.3.2 Feature-Level Fusion Based on Joint Sparse Representation

In the particle filter-based multi-cue tracking framework, we are given \( K \) types of visual cues, e.g. color, shape and texture, to represent the tracking result in the current frame and template images of the target object. Denote the \( k \)-th visual cues of the current tracking result and the \( n \)-th template image as \( y^k \) and \( x^k_n \), respectively. Inspired by the sparse tracking algorithm [87], the tracking result in the current frame can be sparsely represented by a linear combination of the target templates added by an error vector \( \varepsilon^k \) for each visual cue, i.e.

\[
y^k = X^k w^k + \varepsilon^k, k = 1, \cdots, K
\]  

(3.3.3)

where \( w^k \) is a weight vector with dimension \( N \) to reconstruct the current tracking result with visual cue \( y^k \) based on the template set \( X^k = [x^k_1, \ldots, x^k_N] \) and \( N \) is the number of templates.

In (3.3.3), the weight vectors \( w^1, \cdots, w^K \) can be considered as an underlying representation of the tracking result in the current frame with visual cues \( y^1, \cdots, y^K \). In other words, the feature-level fusion is realized by discovering the relationship among the visual cues \( y^1, \cdots, y^K \) to determine weight vectors \( w^1, \cdots, w^K \) dynamically. To learn the optimal fused representation, we define the objective function by minimizing the reconstruction error and a regularization term, i.e.

\[
\min_W \frac{1}{2} \sum_{k=1}^{K} \| y^k - X^k w^k \|_2^2 + \lambda \Omega(W)
\]

(3.3.4)

where \( \| \cdot \|_2 \) represents \( \ell_2 \) norm, \( \lambda \) is a non-negative parameter, \( W = (w^1, \ldots, w^K) \in \mathbb{R}^{N \times K} \) is the matrix of the weight vectors and \( \Omega \) is the regularization function on \( W \).
To derive the regularization function $\Omega$, we assume that the current tracking result can be sparsely represented by the same set of chosen target templates with index $n_1, \ldots, n_c$ for each visual cue, i.e.

$$y^k = w_{n_1}^k x_{n_1}^k + \cdots + w_{n_c}^k x_{n_c}^k + \varepsilon^k, k = 1, \ldots, K \tag{3.3.5}$$

Under the joint sparsity assumption, the number of chosen target templates $c = \|([\|w_1\|_2, \ldots, \|w_N\|_2])_0$ is a small number. Therefore, we can minimize the sparsity measurement as the regularization term in optimization problem (3.3.4). Since the $\ell_0$ norm can be approximated by the $\ell_1$ norm to make the optimization problem tractable, we define $\Omega$ as the following equation similar to that in [142] measuring the class-level sparsity for classification applications,

$$\Omega(W) = \|[\|w_1\|_2, \ldots, \|w_N\|_2]\|_1 = \sum_{n=1}^{N} \|w_n\|_2 \tag{3.3.6}$$

where $w_n$ denotes the $n$-th row in matrix $W$ corresponding to the weights of visual cues for the $n$-th target template. With this formulation, the joint sparsity across different visual cues can be discovered, i.e. $w_n$ becomes zero for a large number of target templates when minimizing the optimization problem (3.3.4). This ensures that all the selected templates (with non-zero weights) play more important roles in reconstructing the current tracking result for all the visual cues.

### 3.3.3 Detecting Unreliable Visual Cues for Robust Fusion

Since some visual cues may be sensitive to some variations, the assumption about shared sparsity may not be valid for tracking. Such unreliable visual cues of the target cannot be sparsely represented by the same set of the selected target templates. That means, for the unreliable visual cue $y^{k'}$, all the target templates are likely to have non-zero weighting for small reconstruction error, i.e.

$$y^{k'} = w_1^{k'} x_1^{k'} + \cdots + w_N^{k'} x_N^{k'} + \varepsilon^{k'} \tag{3.3.7}$$

where $w_1^{k'}, \ldots, w_N^{k'}$ are non-zero weights. In this case, we cannot obtain a robust fusion result by minimizing optimization problem (3.3.4) with the regularization function (3.3.6).
Although unreliable features cannot satisfy (3.3.5), reliable features can still be sparsely represented by (3.3.5) and used to choose the most informative target templates for reconstruction. With the selected templates of index $n_1, \cdots, n_c$, we rewrite (3.3.7) as follows,

$$y_{k'} = \sum_{i=1}^{c} w_{n_i}^{k'} x_{n_i}^{k'} = \sum_{j=1}^{N-c} w_{m_j}^{k'} x_{m_j}^{k'} + \varepsilon^{k'}$$

(3.3.8)

where $m_j$ denotes the index for the template which is not chosen to reconstruct the current tracking result. Suppose we have $K'$ unreliable visual cues. Without loss of generality, let visual cues $1, \cdots, K - K'$ be reliable, while $K - K' + 1, \cdots, K$ be unreliable. To detect the $K'$ unreliable visual cues, we employ the sparsity assumption for the unreliable features, i.e. the number of unreliable visual cues $K' = \| (\sum_{j=1}^{N-c} |w_{m_j}^1|^2, \cdots, \sum_{j=1}^{N-c} |w_{m_j}^K|^2) \|_0$ is a small number, which can be used to define the regularization function. Similar to (3.3.6), the $\ell_1$ norm is used instead of the $\ell_0$ norm. Combining with the regularization function for discovering the joint sparsity among reliable features, the regularization function $\Omega$ in (3.3.4) becomes

$$\Omega(W) = \theta_1 \sum_{n=1}^{N} \sum_{k=1}^{K - K'} |w_{n}^{k}|^2 + \theta_2 \sum_{k=1}^{K} \sum_{j=1}^{N-c} |w_{m_j}^{k}|^2$$

(3.3.9)

where $\theta_1$ and $\theta_2$ are non-negative parameters to balance the joint sparsity across the selected target templates and unreliable visual cues.

However, we have no information about the selected templates and unreliable features before learning, so we cannot a priori define the regularization function like (3.3.9). Inspired by robust multi-task feature learning [32], the weight matrix $W$ can be decomposed into two terms $R$ and $S$ with $W = R + S$. Suppose the non-zero weights of the reliable features be encoded in $R$, while the non-zero weights of the unreliable features encoded in $S$. The current tracking result of the reliable visual cue $k$ can be reconstructed by the information in $R$ only, i.e.(3.3.5) is revised as

$$y_{k} = r_{n_1}^{k} x_{n_1}^{k} + \cdots + r_{n_c}^{k} x_{n_c}^{k} + \varepsilon^{k}, k = 1, \cdots, K - K'$$

(3.3.10)
On the other hand, (3.3.8) for the unreliable feature $k'$ is changed to

$$y^{k'} - \sum_{i=1}^{c} s_{n_i}^{k'} x_{n_i}^{k'} = \sum_{j=1}^{N-c} s_{m_j}^{k'} x_{m_j}^{k'} + \varepsilon^{k'},$$

(3.3.11)

$$k' = K - K' + 1, \cdots, K$$

Based on the above analysis, the final regularization function can be defined analogous to (3.3.9), i.e.

$$\Omega(W) = \theta_1 \sum_{n=1}^{N} \|r_n\|_2 + \theta_2 \sum_{k=1}^{K} \|s^k\|_2$$

(3.3.12)

Denote $\lambda_1 = \lambda \theta_1$ and $\lambda_2 = \lambda \theta_2$. Substituting $\Omega(W)$ in (3.3.12) into optimization problem (3.3.4), the proposed robust joint sparse representation based feature-level fusion tracker (RJSRFFT) is developed as,

$$\min_{W,R,S} \frac{1}{2} \sum_{k=1}^{K} \|y^k - X^k w^k\|_2^2 + \lambda_1 \sum_{n=1}^{N} \|r_n\|_2 + \lambda_2 \sum_{k=1}^{K} \|s^k\|_2$$

s.t. $W = R + S$

(3.3.13)

The procedures to solve the optimization problem (3.3.13) will be given in the following section. The optimal fused representation is given by $R$ and $S$, which encode the sparsity patterns of reliable features and non-sparsity patterns of unreliable visual cues, respectively. Therefore, when the non-sparsity patterns of unreliable features is encoded in $S$, the corresponding columns of coefficients in $S$ would be larger than zero. Therefore, we determine the index set $O$ of the unreliable features as

$$O = \{k', \text{s.t.,} \|s^{k'}\|_2 \geq T\}$$

(3.3.14)

This scheme detects the unreliable visual cues when the norm of some column of matrix $S$ is larger than a pre-defined threshold $T$. That is to say, the features which get strong response in the corresponding columns of $S$ are determined as unreliable features. Since we have no information about the selected templates and unreliable features before learning the reconstruction coefficients, the matrix $S$ may also encode some small components of reconstruction coefficients of reliable features (can be found in Fig.3.1(d) where matrix $S$ shows some small non-zero components of reliable features). Therefore, if the threshold is too small, reliable features which
have small components in S may be mistaken as unreliable features. On the other hand, if it is too large, unreliable features may not be successfully detected. In our experiment, we empirically set it as 0.0007. On the other hand, the likelihood function is defined by \( R \) and \( S \) as follows. The representation coefficients of different visual cues are estimated and the unreliable features are detected by solving the optimization problem (3.3.13). Then, the observation likelihood function in (3.3.2) is defined as

\[
p(z_t | l_t) \propto \exp\left(-\frac{1}{K-K'} \sum_{j \in O} \| y_j^j - X^j \cdot r_j \|_2^2\right) \tag{3.3.15}
\]

where the right side of this equation denotes the average reconstruction error of reliable visual cues. Since the proposed model can detect the unreliable cues, the likelihood function can combine the reconstruction error of reliable cues to define the final similarity between the target candidate and the target templates.

### 3.3.4 Optimization Procedure

The objective function in (3.3.13) consists of a smooth and non-smooth functions. This kind of optimization problem can be solved efficiently by employing Accelerated Proximal Gradient Method (APG). By using the first order information, the APG method can obtain the global optimal solution with the convergence rate \( O\left(\frac{1}{t^2}\right) \), which has been shown in [89]. It has also been further studied in [8][106] and applied in solving multi-task sparse learning/representation problem [143][32][14]. We apply the APG method similar to that in [143] and derive the following algorithm. Let

\[
F(R, S) = \frac{1}{2} \sum_{k=1}^{K} f(r^k, s^k) = \frac{1}{2} \sum_{k=1}^{K} \| y^k - \sum_{n=1}^{N} x_n^k (r_n^k + s_n^k)\|_2^2
\]

\[
G(R, S) = \lambda_1 \sum_{n=1}^{N} \| r_n \|_2 + \lambda_2 \sum_{k=1}^{K} \| s^k \|_2
\]

(3.3.16)

where \( F \) is differentiable convex function with Lipschitz continous gradient, while \( G \) is non-differentiable but convex function. Then by taking the first-order Taylor expansion of \( F \) with the quadratic regularization function, we construct the objective function with the quadratic upper approximation for \( F \) and the non-differentiable
function $G$ at the aggregation matrices $U^t$ and $V^t$ as follows,

$$
\Phi(R,S) = \frac{1}{2} \sum_{k=1}^{K} \{ J(u^{k,t},v^{k,t}) + (\nabla_{u}^{k,t})^T(v^{k} - u^{k,t}) \\
+ (\nabla_{v}^{k,t})^T(s^{k} - v^{k,t}) \} + \frac{\mu}{2} ||r^{k} - u^{k,t}||^2_2 + \frac{\mu}{2} ||s^{k} - v^{k,t}||^2_2 \\
+ \lambda_1 \sum_{n=1}^{N} ||r_n||_2 + \lambda_2 \sum_{k=1}^{K} ||s_k||_2
$$

(3.3.17)

where $\mu$ is the Lipschitz constant [8], $\nabla_{u}^{k,t}$ and $\nabla_{v}^{k,t}$ are the gradient operator of $F(U,V)$ on $u^{k,t}$ and $v^{k,t}$, respectively. In the $(t+1)$-th iteration, given the aggregation matrices $U^t$ and $V^t$, the proximal matrices $R^{t+1}$ and $S^{t+1}$ are obtained by minimizing the following optimization problem,

$$
\arg\min_{R,S} \Phi(R,S)
$$

(3.3.18)

With some algebraic manipulations, (3.3.18) can be separated into two independent sub-problems about $R$ and $S$, respectively, i.e.

$$
\min_{R} \frac{1}{2} \sum_{k=1}^{K} ||r^{k} - (u^{k,t} - \frac{1}{\mu} \nabla_{u}^{k,t})||^2_2 + \frac{\lambda_1}{\mu} \sum_{n=1}^{N} ||r_n||_2
$$

$$
\min_{S} \frac{1}{2} \sum_{k=1}^{K} ||s^{k} - (v^{k,t} - \frac{1}{\mu} \nabla_{v}^{k,t})||^2_2 + \frac{\lambda_2}{\mu} \sum_{k=1}^{K} ||s_k||_2
$$

(3.3.19)

where the gradient operators are given by $\nabla_{u}^{k,t} = -(X^{k})^T y^{k} + (X^{k})^T(X^{k})u^{k,t} + (X^{k})^T(X^{k})v^{k,t}$, $\nabla_{v}^{k,t} = -(X^{k})^T y^{k} + (X^{k})^T(X^{k})u^{k,t} + (X^{k})^T(X^{k})v^{k,t}$. As in [143], we solve the above subproblem using the following two steps iteratively:

**Gradient Mapping Step** the proximal matrices $R^{t+1}$ and $S^{t+1}$ are updated by (3.3.20) and (3.3.21), respectively.

$$
r^{k,t+\frac{1}{2}} = u^{k,t} - \frac{1}{\mu} \nabla_{u}^{k,t}, k = 1, \cdots, K,
$$

$$
r^{t+1}_n = \max(0,1 - \frac{\lambda_1}{\mu ||r^{t+\frac{1}{2}}||_2}) \cdot r^{t+\frac{1}{2}}_n, n = 1, \cdots, N
$$

(3.3.20)

$$
s^{k,t+\frac{1}{2}} = v^{k,t} - \frac{1}{\mu} \nabla_{v}^{k,t}, k = 1, \cdots, K,
$$

$$
s^{k,t+1} = \max(0,1 - \frac{\lambda_2}{\mu ||s^{k,t+\frac{1}{2}}||_2}) \cdot s^{k,t+\frac{1}{2}}, k = 1, \cdots, K
$$

(3.3.21)
Algorithm 3: Optimization Procedure for problem (3.3.13)

Input: Template set \( \{X^k\}_{k=1}^K \), target candidate sample \( \{y^k\}_{k=1}^K \), regularization parameters \( \lambda_1 \) and \( \lambda_2 \), Lipschitz constant \( \mu \)

Output: Weight matrix \( R \) and \( S \)

Initialization: \( U^0 \leftarrow 0, V^0 \leftarrow 0, G_{u0}^{k,0} \leftarrow 0, G_{v0}^{k,0} \leftarrow 0 \), and \( a_0 \leftarrow 1 \)

repeat
  \[
  G_{u}^{k,t} = (X^k)^T (X^k) u^{k,t} + (X^k)^T (X^k) v^{k,t} - (X^k)^T y^k, \quad k = 1, \ldots, K
  \]
  \[
  G_{v}^{k,t} = (X^k)^T (X^k) v^{k,t} + (X^k)^T (X^k) u^{k,t} - (X^k)^T y^k, \quad k = 1, \ldots, K
  \]
  \[
  r^{k,t+\frac{1}{2}} = u^{k,t} - \frac{1}{\mu} G_{u}^{k,t}, \quad k = 1, \ldots, K
  \]
  \[
  r^{n,t+1} = \max(0, 1 - \frac{\lambda_1}{\mu \| r^{n,t+\frac{1}{2}} \|_2}), \quad n = 1, \ldots, N
  \]
  \[
  s^{k,t+\frac{1}{2}} = v^{k,t} - \frac{1}{\mu} G_{v}^{k,t}, \quad k = 1, \ldots, K
  \]
  \[
  s^{k,t+1} = \max(0, 1 - \frac{\lambda_2}{\mu \| s^{k,t+\frac{1}{2}} \|_2}), \quad k = 1, \ldots, K
  \]
  \[
  a_{t+1} = \frac{1+\sqrt{1+4a_t^2}}{2}
  \]
  \[
  U^{t+1} = R^{t+1} + \frac{a_t-1}{a_{t+1}} (R^{t+1} - R^t)
  \]
  \[
  V^{k+1} = S^{k+1} + \frac{a_t-1}{a_{t+1}} (S^{k+1} - S^t)
  \]
until convergence

It should be noted that the update schemes (3.3.20) for \( R \) and (3.3.21) for \( S \) are different from each other, since \( R \) and \( S \) have different sparsity properties grouping according to columns and rows, respectively.

Aggregation Step  the aggregation matrix is updated as follows.

\[
U^{t+1} = R^{t+1} + \frac{a_t-1}{a_{t+1}} (R^{t+1} - R^t), \quad S^{k+1} = S^{k+1} + \frac{a_t-1}{a_{t+1}} (S^{k+1} - S^t)
\]

where \( a_{t+1} = \frac{1+\sqrt{1+4a_t^2}}{2} \), and \( a_0 = 1 \).

The overall optimization procedure is summarized in Algorithm 3.
Algorithm 4: Optimization Procedure for problem (3.3.25)

**Input:** \{\mathcal{K}(X^k, X^k)\}_{k=1}^{K} \text{ and } \{\mathcal{K}(X^k, y^k)\}_{k=1}^{K}, \text{ regularization parameters } \lambda_1 \text{ and } \lambda_2, \text{ Lipschitz constant } \mu

**Output:** Weight matrix \( R \) and \( S \)

**Initialization:** \( U^0 \leftarrow 0, V^0 \leftarrow 0, G_u^{k,0} \leftarrow 0, G_v^{k,0} \leftarrow 0, \text{ and } a_0 \leftarrow 1 \)

repeat

\[
G_u^{k,t} = \mathcal{K}(X^k, X^k) u^{k,t} + \mathcal{K}(X^k, X^k) v^{k,t} - \mathcal{K}(X^k, y^k), \quad k = 1, \ldots, K
\]

\[
G_v^{k,t} = \mathcal{K}(X^k, X^k) v^{k,t} + \mathcal{K}(X^k, X^k) u^{k,t} - \mathcal{K}(X^k, y^k), \quad k = 1, \ldots, K
\]

\[
r_{t+\frac{1}{2}}^{k,t} = u^{k,t} - \frac{1}{\mu} G_u^{k,t}, \quad k = 1, \ldots, K, \quad r_{t+1} = \max(0, 1 - \frac{\lambda_1}{\mu \| r_{t+\frac{1}{2}} \|_2}) \cdot r_{t+\frac{1}{2}},
\]

\[
s_{t+\frac{1}{2}}^{k,t} = v^{k,t} - \frac{1}{\mu} G_v^{k,t}, \quad k = 1, \ldots, K
\]

\[
s_{t+1}^{k,t} = \max(0, 1 - \frac{\lambda_2}{\mu \| s_{t+\frac{1}{2}} \|_2}) \cdot s_{t+\frac{1}{2}}, \quad k = 1, \ldots, K.
\]

\[
U^{t+1} = R^{t+1} + \frac{a_t - 1}{a_{t+1}} (R^{t+1} - R^t)
\]

\[
V^{t+1} = S^{t+1} + \frac{a_t - 1}{a_{t+1}} (S^{t+1} - S^t)
\]

until convergence

3.3.5 Kernelized Framework for Robust Feature-Level Fusion

In order to exploit the non-linearity of features which has been shown to enhance the performance in many computer vision tasks [135] [98] [28], we extend our algorithm into a general framework that can combine multiple visual features from different kernel space while detecting unreliable features for tracking. Let \( \phi \) denote the mapping function to a kernel space. According to (3.3.3), sparse representation with template set for tracking result in linear space can be extended to a kernel space as follows,

\[
\phi(y^k) = \Phi(X^k) w^k + \epsilon^k, \quad k = 1, \ldots, K.
\]  

(3.3.23)

where \( \Phi(X^k) = [\phi(x_1^k), \cdots, \phi(x_N^k)] \) denotes the template set in kernel space via the mapping function \( \phi \). Based on the derivation leading to (3.3.13), the robust joint kernel sparse representation based feature-level fusion (K-JSRFFT) model for
visual tracking is developed as,
\[
\min_{W,R,S} \frac{1}{2} \sum_{k=1}^{K} \| \phi(y^k) - \Phi(X^k)w^k \|^2_2 + \lambda_1 \sum_{n=1}^{N} \| r_n \|^2_2 + \lambda_2 \sum_{k=1}^{K} \| s^k \|^2_2
\]
\[\text{s.t. } W = R + S \] (3.3.24)

Expanding the reconstruction term in optimization problem (3.3.24), we can reformulate problem (3.3.24) as,
\[
\min_{W,R,S} \frac{1}{2} \sum_{k=1}^{K} \{ \mathcal{K}(y^k, y^k) - 2 \cdot (w^k)^T \mathcal{K}(X^k, y^k) \\
+ (w^k)^T \mathcal{K}(X^k, X^k)w^k \} + \lambda_1 \sum_{n=1}^{N} \| r_n \|^2_2 + \lambda_2 \sum_{k=1}^{K} \| s^k \|^2_2
\]
\[\text{s.t. } W = R + S \] (3.3.25)

Here the element in the $i$-th row and $j$-th columns of the kernel matrix $\mathcal{K}(X,Y)$ is defined as,
\[
\mathcal{K}_{i,j}(X,Y) = (\phi(x_i))^T \phi(y_j)
\] (3.3.26)

where $x_i$ and $y_j$ are the $i$-th and $j$-th column of $X$ and $Y$, respectively. And, $\mathcal{K}(y^k, y^k)$ in problem (3.3.25) is equal to 1 because of the normalization to unit length. Based on Algorithm 3, the optimization procedure for problem (3.3.25) can be derived and is summarized in Algorithm 4.

After solving (3.3.25), the representation coefficients of different visual cues are estimated and the unreliable features in kernel space are detected. Then, the observation likelihood function can be re-defined as,
\[
p(z_t|l_t) \propto \exp\left( -\frac{1}{K - K'} \sum_{j \notin O} [\mathcal{K}(y^j, y^j) - 2 \cdot (w^j)^T \mathcal{K}(X^j, y^j) \\
+ (w^j)^T \mathcal{K}(X^j, X^j)w^j] \right)
\] (3.3.27)

where $O$ denotes the index set of unreliable features in kernel space as in (3.3.14), $K$ and $K'$ denotes the total number of features and unreliable features in kernel space, respectively.

### 3.3.6 Computational Complexity

The major computation time of the proposed tracker is due to the following processes: feature extraction, kernel matrix computation, Lipschitz constant estimation
and the optimization procedure. The linear/non-linear kernel matrix for the feature
dictionary is computed once the template update scheme is performed. Therefore,
assuming that the flop count of the kernel mapping of two feature vectors is \( p \), the
computational complexity of the kernel matrix computation for the corresponding
feature dictionary is \( O(N^2p) \) where \( N \) is the number of templates in the template
set. By using the tight Lipschitz constant estimation as discussed in Section 3.4,
the computational complexity for Lipschitz constant estimation is reduced from
\( O(K^3N^3) \) to \( O(N^3) \), where \( K \) is the number of visual cues. The optimization proce-
dure as illustrated in Algorithms 1 and 2 is dominated by the gradient computation
in steps 3 and 4. The complexity of Steps 3 and 4 are \( O(KN^2n) \) where \( n \) is the
number of particles. The proposed tracker is implemented in MATLAB without
code optimization.

As measured on an i7 Quad-Core machine with 3.4 GHz CPU, the running time
of K-RJSRFFT and RJSRFFT are 2 sec/frame and 1.5 sec/frame, respectively. We
will explore ways to increase its computation efficiency in our future work so as to
perform in real-time.

## 3.4 Implementation Details

### 3.4.1 Template Update Scheme

The proposed tracker is sparse-based. We adopt the template update scheme in [87]
with a minor modification to fit our proposed fusion-based tracker with an outlier
detection scheme. Similar to [87], we associate each template in different visual
cues with a weight, and the weight is updated in each frame. Once the similarity
between the templates with the largest weight from the reliable visual cue and the
target sample of the corresponding visual cue is less than a predefine threshold, the
proposed tracker will replace the template which has the least weight with the target
sample. The difference between [87] and the proposed method is that the update
scheme in the proposed method is performed simultaneously for template sets in
different visual cues. Once one of the templates in a visual cue is replaced, the
template in other visual cues will be replaced because the proposed model performs
multi-cue fusion at feature level. As such, all the cues of the same template should be
updated simultaneously. The template update scheme is summarized in Algorithm
5.

It should be noted that the similarity threshold determines the template update
frequency. A higher similarity threshold would result in a higher updating frequency
while a lower similarity threshold would lead to a lower updating frequency. If the
template set is updated too frequently, small errors would be accumulated and the
tracker would drift from the target gradually. On the other hand, if the template
set remains unchanged for a long time, the template set may not be well adaptive
to appearance and background change. In our experiments, we set the similarity
threshold as $\cos(35^\circ) \approx 0.82$.

### 3.4.2 Tight Lipschitz Constant Estimation

The Lipschitz constant $\mu$ is important to above optimization algorithm. An improper
value of $\mu$ will result in either divergence or slow convergence in above optimization
algorithm. In this subsection, we will derive the method for tight Lipschitz constant
estimation. As such, the Lipschitz constant will be automatically set in the initial
frame or update once the template update scheme is performed. Proposition 1 gives
the method for tight Lipschitz constant estimation.

**Proposition 3.4.1.** Let $F$ denote the function defined in (3.3.16) with $\{X_k|k = 1,\ldots,K\}$, where $X_k$ denote the template set in the $k$-th visual cue. The Lipschitz
constant for $\nabla F$ can be estimated via its tight lower bound as follows,

$$
\mu \geq \max\{2 \cdot \lambda_{\max}^k|k = 1,\ldots,K\}
$$

(3.4.28)

where $\lambda_{\max}^k$ is the largest eigenvalue of $K(X_k, X_k)$.

The proof of Proposition 1 can be found in Appendix A. We can see that with
this method in (3.4.28), the tight lower bound of the Lipschitz can be estimated just
**Algorithm 5: Template Update Scheme**

**Input:**

- $X^k, k = 1, ..., K$: the object template set of multiple features
- $y^k, k = 1, ..., K$: the current tracking result of multiple features
- $\omega^k_n, k = 1, ..., K - K', n = 1, ..., N$: the weights associated with each template of each reliable feature.
- $R^k, k = 1, ..., K - K'$: the reconstruction coefficients of each template of each reliable feature.
- $\tau$: Similarity threshold.

**Output:** $\omega^k_n, k = 1, ..., K - K', n = 1, ..., N, X^k, k = 1, ..., K - K', n = 1, ..., N$

// **Updating Importance Weight:**

$\omega^k_n \leftarrow \omega^k_n \cdot \exp(r^k_n), k = 1, ..., K - K'$

// **Selecting the template with maximum weight:**

$(n', k') \leftarrow \text{arg max}_{n,k} \omega^k_n$

// **Similarity computation:**

$\theta \leftarrow \text{sim}(y^{k'}, x^{k'}_{n'})$

// **Selecting the template with maximum weight:**

$(n', k') \leftarrow \text{arg max}_{n,k} \omega^k_n$

// **Similarity computation:** $\theta \leftarrow \text{sim}(y^{k'}, x^{k'}_{n'})$

if $\theta < \tau$ then

// **Selecting the template with minimum weight:**

$(\hat{n}, \hat{k}) \leftarrow \text{arg min}_{n,k} \omega^k_n$

// **Updating templates:**

$x^k_{\hat{n}} \leftarrow y^k_{\hat{n}}, k = 1, ..., K - K'$

// **Assigning weight:**

$\omega^k_{\hat{n}} = \text{median}(\omega^k), k = 1, ..., K - K'$

Normalize $\omega^k, k = 1, ..., K - K'$ such that $\|\omega^k\|_2^2 = 1$.

end
via the maximum eigenvalues of the sub-blocks of the Hessian matrix. Therefore, it avoids the direct computation of the eigenvalues of the Hessian matrix in high dimension. In the experiment, we set the value of Lipschitz constant as the lower bound.

3.5 Experiments

In this section, we evaluate the proposed robust joint sparse representation based feature-level fusion tracker (RJSRFFT) and its kernelized method (K-RJSRFFT) using both synthetic data and real videos from publicly available datasets.

3.5.1 Unreliable Feature Detection on Synthetic Data

To demonstrate that the proposed method can detect unreliable features, we compare the RJSRFFT with the weight matrices obtained by solving (3.3.4) with the regularization term (3.3.6) as in the multi-task joint sparse representation (MTJSR) method [142]. In this experiment, we simulate the multi-cue tracking problem by randomly generating five kinds of ten dimensional normalized features with 30 templates, i.e. $X^k \in \mathbb{R}^{10 \times 30}, k = 1, \cdots, 5$ are the template sets. Two kinds of features are set as unreliable with non-sparsity patterns. For the other three kinds of reliable features, we divide the template sets into three groups and randomly generate the template weight vector $w^k \in \mathbb{R}^{30}$, such that the elements in $w^k$ corresponding to only one group of templates are non-zero. The testing sample of the $k$-th feature $y^k$ to represent the current tracking result is computed by $X^k w^k$ plus a Gaussian noise vector with zero mean and variance 0.2 to represent the reconstruction error $\varepsilon^k$. For a fair comparison with the MTJSR [142], we extend our model to impose the group lasso penalty by simply using a group sparsity term in optimization problem (3.3.13). We empirically set parameters $\lambda, \lambda_1, \lambda_2$ as 0.001 and the step size $\mu$ as 0.002 and repeat the experiment 100 times.

We use the average normalized mean square error between the original weight
matrix and the recovered one for evaluation. Our method RJSRFFT achieves a much lower average recovery error of 4.69% compared with that of the MTJSR with 12.29%. This indicates that our method can better recover the underlying weight matrix by detecting the unreliable features successfully. To further demonstrate the ability for unreliable feature detection, we give a graphical illustration of one out of the 100 experiments in Fig.3.1. The original weight matrix is shown in Fig.3.1(a) with each row representing a weight vector $w^k$. The horizontal axis records the sample indexes, while the vertical gives the values of weights. From Fig.3.1(a), we can see that the first three share the same sparsity patterns over the samples with indexes in the middle range, while all the weights of the last two features are non-zeros, thus non-sparse. In this case, the MTJSR cannot discover the sparsity patterns as shown in Fig.3.1(b), while the proposed RJSRFFT can find out the shared sparsity of the reliable features and detect unreliable features as shown in Fig.3.1(c) and 3.1(d). This also explains the reason why our method can better recover the underlying matrix as shown in Fig.3.1(a).

### 3.5.2 Visual Tracking Experiments

To evaluate the performance of the proposed tracking algorithms, i.e., RJSRFFT and K-RJSRFFT, we conduct experiments on thirty-three publicly available video sequences\(^1\). The challenging factors in these video sequences include occlusion, cluttered background, change of scale, illumination and pose, which are listed in Table 3.1. We compare the proposed tracking algorithms, RJSRFFT and K-RJSRFFT with state-of-the-art tracking methods including multi-cue tracker: online boosting (OAB) [33], online multiple instance learning (MIL) [4], semi-supervised online

Figure 3.1: Graphical illustration of unreliable feature detection on synthetic data
Figure 3.2: Graphical Illustration of Unreliable Feature Detection Scheme. The upper-left image in purple box show the tracking result in the 658-th frame of video *Occluded Face 1*. The seven figures in the upper-right green box illustrate the resulting reconstruction coefficients via solving the $\ell_1$ minimization problem as shown in small green box. After solving (3.3.24), the resulting reconstruction coefficients of each feature encoded in matrix $R$ are illustrated in the bottom-left blue box, and the unreliable detection results encoded in matrix $S$ are illustrated in the bottom-right blue box. The $y$-coordinate of all above figures denotes the value of coefficient, and the $x$-coordinate denotes the index of the template in the template set.
Table 3.1: The Challenging Factor and Its Corresponding Videos

<table>
<thead>
<tr>
<th>Challenging Factor</th>
<th>Video Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occlusion</td>
<td>Occluded Face 1, Occluded Face 2, Football, Skating1, CarScale, David Indoor, David Outdoor, Basketball, Bolt, Dudek, Subway, Walking, Walking2</td>
</tr>
<tr>
<td>Cluttered Background</td>
<td>Crossing, Mountain-bike, Animal, Football, Football1, Skating1, Car11, Shaking, David Outdoor, Basketball, Dudek, Jumping, Singer2, Subway</td>
</tr>
<tr>
<td>Scale</td>
<td>Crossing, Trellis, Skating1, CarScale, Car4, David Indoor, Dog, Dudek, FleetFace, Fleetman1, Freeman3, Singer1, walking, Walking2,</td>
</tr>
<tr>
<td>Illumination</td>
<td>Trellis, Skating1, Car11, Car4, David Indoor, Shaking, Basketball, Fish, Singer1, Singer2, Sylvester</td>
</tr>
<tr>
<td>Pose</td>
<td>Trellis, Mountain-bike, Occluded Face 2, Football, Football1, Skating1, David Indoor, David, David Outdoor, Shaking, Basketball, Bolt, Dog, Dudek, FleetFace, Freeman1, Freeman2, Singer2, Sylvester, Ming-Hsuan, Boy</td>
</tr>
</tbody>
</table>

boosting (SemiB) [34], sparse representation-based tracker: \( \ell_1 \) tracker (L1APG) [7], sparse collaborative model (SCM) [162], multi-task tracker (MTT) [158], and other state-of-the-art methods: struck method (Struck) [37], circulant structure tracker (CSK) [40], fragment tracker (Frag) [1], incremental learning tracker (IVT) [93], distribution field tracker (DFT) [95], compressive tracker (CT) [149]. The source codes of these trackers are provided by the authors of these papers. For a fair comparison, all the trackers are set to be with the same initialization parameters. To illustrate the effectiveness of the proposed unreliable feature detection scheme that encodes the non-zero weights of the reliable feature in matrix \( S \) and detect the unreliable feature based on the vector norm, we implement the Joint Sparse Representation based Feature-level Fusion Tracker without unreliable feature detection scheme (JSRFFT) for short as discussed in Section 3.3.2. Both qualitative and quantitative evaluation are presented in 3.5.2.
Experiment Settings

For the proposed K-RJSRFFT, we extract six kinds of local and global features with totally 3 kinds of kernel for fusion from a normalized 32 by 32 image patch representing the target observation. For local visual cues, we divide the tracking bounding box into 2 by 2 blocks and extract covariance descriptor in each block with log-Euclidean kernel function as in [70]. For global visual cues, we use HOG [20] with linear and polynomial kernel, and GLF [168] with linear kernel. For RJSRFFT and JSRFFT, we extract the same kinds of feature but only with linear kernel. We empirically set the parameters as follows. $\lambda_1$ and $\lambda_2$ is set to 0.57 and 0.87, respectively. The threshold $T$ in (3.3.14) is 0.0007. And the number of templates is 12, and 200 samples are drawn for particle filtering. We adopt the cosine function as the similarity function, and empirically set the similarity threshold $\tau$ as $\cos(35^\circ) \approx 0.82$.

Visualization Experiment

To illustrate how unreliable features are detected in the proposed method, for the tracking result in the bounding box as shown in the upper-left image in Fig. 3.2, we use features of the tracking results and the template set to obtain the reconstruction coefficients via solving the kernel sparse representation problem for each feature as shown in the small green box, and plot the value of the coefficients of each template of all features. The higher value of coefficient means the corresponding template is more representative for reconstructing the target. As we can see, the reconstruction coefficients of GLF features do not demonstrate sparsity pattern as other features, which means the GLF features extracted from the tracking result cannot be well represented by the corresponding template set. We can also note that for most features, the sixth template is selected as the most representative template with the highest value of coefficient, which implies the underlying relationship between different features. Through our proposed model, the sixth template are jointly selected as the representative template for target representation as shown in the
bottom-left figure and the GLF features are also detected as an unreliable feature as shown in the bottom-right figure.
Table 3.2: Center Location Error (in pixels). The best three results are shown in red, blue and green.

(a)

<table>
<thead>
<tr>
<th></th>
<th>Crossing</th>
<th>Trellis</th>
<th>MountainBike</th>
<th>FaceOcc1</th>
<th>FaceOcc2</th>
<th>Deer</th>
<th>Football</th>
<th>Football1</th>
<th>Skating1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>3</td>
<td>7.1</td>
<td>9</td>
<td>19.3</td>
<td>6</td>
<td>5.3</td>
<td>17.2</td>
<td>5.3</td>
<td>83.2</td>
</tr>
<tr>
<td>L1APG</td>
<td>62.2</td>
<td>62.3</td>
<td>8.5</td>
<td>17.8</td>
<td>12.9</td>
<td>24.1</td>
<td>15</td>
<td>9.8</td>
<td>159.2</td>
</tr>
<tr>
<td>SCM</td>
<td>2</td>
<td>6.9</td>
<td>10.8</td>
<td>13.6</td>
<td>9</td>
<td>103.7</td>
<td>16.2</td>
<td>20.7</td>
<td>16.5</td>
</tr>
<tr>
<td>CSK</td>
<td>8.3</td>
<td>19.2</td>
<td>6.4</td>
<td>12.5</td>
<td>6</td>
<td>4.8</td>
<td>16</td>
<td>17.2</td>
<td>8</td>
</tr>
<tr>
<td>Frag</td>
<td>38.3</td>
<td>59.6</td>
<td>206.3</td>
<td>11.5</td>
<td>15.9</td>
<td>104.7</td>
<td>5.5</td>
<td>15.5</td>
<td>149.6</td>
</tr>
<tr>
<td>MIL</td>
<td>2.9</td>
<td>72</td>
<td>72.6</td>
<td>30.1</td>
<td>13.9</td>
<td>100.2</td>
<td>12.2</td>
<td>6.2</td>
<td>139.6</td>
</tr>
<tr>
<td>OAB</td>
<td>4.6</td>
<td>98.7</td>
<td>12.2</td>
<td>25.1</td>
<td>19.8</td>
<td>6.8</td>
<td>72.6</td>
<td>85.4</td>
<td>43.2</td>
</tr>
<tr>
<td>MTT</td>
<td>56.5</td>
<td>69</td>
<td>7.3</td>
<td>21.6</td>
<td>10.2</td>
<td>19.1</td>
<td>13.4</td>
<td>13.1</td>
<td>293.8</td>
</tr>
<tr>
<td>IVT</td>
<td>2</td>
<td>120</td>
<td>8.1</td>
<td>18.9</td>
<td>7.8</td>
<td>182.6</td>
<td>14.2</td>
<td>24.8</td>
<td>247.6</td>
</tr>
<tr>
<td>DFT</td>
<td>21.8</td>
<td>44.3</td>
<td>154.7</td>
<td>23.8</td>
<td>8.1</td>
<td>98.3</td>
<td>9.2</td>
<td>2.6</td>
<td>174.6</td>
</tr>
<tr>
<td>SemiB</td>
<td>3.8</td>
<td>69.5</td>
<td>249.7</td>
<td>75.1</td>
<td>50.9</td>
<td>61.6</td>
<td>244.7</td>
<td>162.6</td>
<td>288.3</td>
</tr>
<tr>
<td>CT</td>
<td>3.2</td>
<td>42.2</td>
<td>213.7</td>
<td>26.3</td>
<td>18.9</td>
<td>246.2</td>
<td>11.8</td>
<td>20.1</td>
<td>150.9</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>2.1</td>
<td>5.9</td>
<td>164.7</td>
<td>24.2</td>
<td>10.3</td>
<td>84.3</td>
<td>13.1</td>
<td>14.9</td>
<td>136.6</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>1.9</td>
<td>4.3</td>
<td>61.2</td>
<td>12.6</td>
<td>8.2</td>
<td>27.9</td>
<td>14.2</td>
<td>3.7</td>
<td>13.2</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>2.9</td>
<td>6.5</td>
<td>10.4</td>
<td>13.7</td>
<td>6.5</td>
<td>5.4</td>
<td>5.2</td>
<td>4.4</td>
<td>7.1</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th></th>
<th>CarDark</th>
<th>CarScale</th>
<th>Car4</th>
<th>David</th>
<th>David2</th>
<th>David3</th>
<th>Shaking</th>
<th>Basketball</th>
<th>Bolt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>1.1</td>
<td>36.2</td>
<td>8.8</td>
<td>42.3</td>
<td>3.1</td>
<td>106.7</td>
<td>30.1</td>
<td>118.4</td>
<td>399.3</td>
</tr>
<tr>
<td>L1APG</td>
<td>1.4</td>
<td>80.2</td>
<td>77</td>
<td>13.8</td>
<td>1.3</td>
<td>89.8</td>
<td>109.7</td>
<td>137.7</td>
<td>409</td>
</tr>
<tr>
<td>SCM</td>
<td>1.9</td>
<td>33.8</td>
<td>4.6</td>
<td>4.1</td>
<td>3.1</td>
<td>73.4</td>
<td>11.2</td>
<td>52.8</td>
<td>456.9</td>
</tr>
<tr>
<td>CSK</td>
<td>3.8</td>
<td>83.2</td>
<td>19.5</td>
<td>17.8</td>
<td>1.9</td>
<td>56.2</td>
<td>17.6</td>
<td>6.6</td>
<td>430</td>
</tr>
<tr>
<td>Frag</td>
<td>36.7</td>
<td>20</td>
<td>131.6</td>
<td>82.5</td>
<td>56.9</td>
<td>13.6</td>
<td>192</td>
<td>12.9</td>
<td>183.5</td>
</tr>
<tr>
<td>MIL</td>
<td>44</td>
<td>34.1</td>
<td>50.2</td>
<td>17.3</td>
<td>11.5</td>
<td>30</td>
<td>24</td>
<td>92.2</td>
<td>394.1</td>
</tr>
<tr>
<td>OAB</td>
<td>2.9</td>
<td>30.2</td>
<td>95.9</td>
<td>22.2</td>
<td>33.6</td>
<td>83.5</td>
<td>191.6</td>
<td>205.5</td>
<td>256.5</td>
</tr>
<tr>
<td>MTT</td>
<td>1.8</td>
<td>87.8</td>
<td>22.6</td>
<td>32.5</td>
<td>1.3</td>
<td>341.8</td>
<td>97.3</td>
<td>106.9</td>
<td>409.2</td>
</tr>
<tr>
<td>IVT</td>
<td>8.9</td>
<td>12.2</td>
<td>2.4</td>
<td>4.6</td>
<td>1.2</td>
<td>51.9</td>
<td>86.1</td>
<td>107.3</td>
<td>397.5</td>
</tr>
<tr>
<td>DFT</td>
<td>59</td>
<td>76.1</td>
<td>62.5</td>
<td>42.5</td>
<td>17.2</td>
<td>51</td>
<td>263</td>
<td>18.4</td>
<td>367.7</td>
</tr>
<tr>
<td>SemiB</td>
<td>2.5</td>
<td>132.1</td>
<td>152.7</td>
<td>46.8</td>
<td>11.8</td>
<td>237</td>
<td>275</td>
<td>328.4</td>
<td>363.7</td>
</tr>
<tr>
<td>CT</td>
<td>119.3</td>
<td>25.9</td>
<td>86</td>
<td>10.9</td>
<td>77.1</td>
<td>88.6</td>
<td>79.4</td>
<td>89.5</td>
<td>364.2</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>3.2</td>
<td>70.1</td>
<td>9.8</td>
<td>4.4</td>
<td>2.5</td>
<td>52.4</td>
<td>65.1</td>
<td>265.2</td>
<td>395.6</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>2.6</td>
<td>22.5</td>
<td>3.3</td>
<td>8.3</td>
<td>2.3</td>
<td>7.5</td>
<td>35.3</td>
<td>227.3</td>
<td>361.7</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>3</td>
<td>25.2</td>
<td>3.9</td>
<td>5.1</td>
<td>1.5</td>
<td>7.1</td>
<td>8.2</td>
<td>15.4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Dog1</td>
<td>Dudek</td>
<td>Fish</td>
<td>FleetFace</td>
<td>Freeman1</td>
<td>Freeman3</td>
<td>Jumping</td>
<td>Singer1</td>
<td>Singer2</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>Struck</td>
<td>5.6</td>
<td>11.8</td>
<td>3.5</td>
<td>22.8</td>
<td>14.2</td>
<td>17.1</td>
<td>7.2</td>
<td>14.7</td>
<td>174.8</td>
</tr>
<tr>
<td>L1APG</td>
<td>3.6</td>
<td>23.4</td>
<td>29.1</td>
<td>113.9</td>
<td>62.2</td>
<td>33.5</td>
<td>83.3</td>
<td>53.2</td>
<td>181.3</td>
</tr>
<tr>
<td>SCM</td>
<td>7</td>
<td>10.9</td>
<td>8.9</td>
<td>27.5</td>
<td>6.9</td>
<td>3.6</td>
<td>65.5</td>
<td>2.6</td>
<td>113.9</td>
</tr>
<tr>
<td>CSK</td>
<td>4.1</td>
<td>13.5</td>
<td>40.9</td>
<td>25.5</td>
<td>125.6</td>
<td>54.2</td>
<td>85.3</td>
<td>14.4</td>
<td>185.7</td>
</tr>
<tr>
<td>Frag</td>
<td>11.7</td>
<td>82.9</td>
<td>21.3</td>
<td>67.3</td>
<td>10.1</td>
<td>40.6</td>
<td>5.8</td>
<td>89</td>
<td>88.9</td>
</tr>
<tr>
<td>MIL</td>
<td>7.5</td>
<td>17.8</td>
<td>23.6</td>
<td>62.7</td>
<td>10.8</td>
<td>87.4</td>
<td>10.6</td>
<td>16.6</td>
<td>22.9</td>
</tr>
<tr>
<td>OAB</td>
<td>5.9</td>
<td>31.7</td>
<td>86.5</td>
<td>51.8</td>
<td>35.8</td>
<td>40.9</td>
<td>46.5</td>
<td>12.6</td>
<td>188.1</td>
</tr>
<tr>
<td>MTT</td>
<td>3.9</td>
<td>14.3</td>
<td>45.1</td>
<td>68.9</td>
<td>117.9</td>
<td>15.7</td>
<td>84.1</td>
<td>36.1</td>
<td>210.1</td>
</tr>
<tr>
<td>IVT</td>
<td>3.6</td>
<td>9.8</td>
<td>6.3</td>
<td>62</td>
<td>11.8</td>
<td>35.8</td>
<td>61.8</td>
<td>10.9</td>
<td>175.9</td>
</tr>
<tr>
<td>DFT</td>
<td>41.4</td>
<td>18.8</td>
<td>9.1</td>
<td>67.9</td>
<td>10.7</td>
<td>32.4</td>
<td>67</td>
<td>19.3</td>
<td>21.4</td>
</tr>
<tr>
<td>SemiB</td>
<td>75.7</td>
<td>218.9</td>
<td>62.7</td>
<td>262.8</td>
<td>175</td>
<td>153.3</td>
<td>94.3</td>
<td>318.9</td>
<td>.356.3</td>
</tr>
<tr>
<td>CT</td>
<td>6.6</td>
<td>26.9</td>
<td>10.7</td>
<td>57.8</td>
<td>118.9</td>
<td>65</td>
<td>47.2</td>
<td>15.9</td>
<td>127.4</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>10.6</td>
<td>16.6</td>
<td>11</td>
<td>68.5</td>
<td>9.2</td>
<td>4</td>
<td>5.7</td>
<td>8.5</td>
<td>61.4</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>5</td>
<td>8.5</td>
<td>7.6</td>
<td>18.7</td>
<td>5.6</td>
<td>3</td>
<td>7</td>
<td>2.7</td>
<td>17.7</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>3.5</td>
<td>9.9</td>
<td>4.1</td>
<td>16.9</td>
<td>7.3</td>
<td>3</td>
<td>4.4</td>
<td>5</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Subway</th>
<th>Walking</th>
<th>Walking2</th>
<th>Sylvester</th>
<th>Mhyang</th>
<th>Boy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>4.1</td>
<td>4.3</td>
<td>10.7</td>
<td></td>
<td>6</td>
<td>2.6</td>
</tr>
<tr>
<td>L1APG</td>
<td>148.1</td>
<td>2.9</td>
<td>4.5</td>
<td>25.9</td>
<td>3.3</td>
<td>7</td>
</tr>
<tr>
<td>SCM</td>
<td>3.4</td>
<td>2.7</td>
<td>1.5</td>
<td>7.8</td>
<td>2.8</td>
<td>51.1</td>
</tr>
<tr>
<td>CSK</td>
<td>164.8</td>
<td>6.5</td>
<td>17.3</td>
<td>10.1</td>
<td>3.4</td>
<td>20.1</td>
</tr>
<tr>
<td>Frag</td>
<td>16.1</td>
<td>9</td>
<td>57.1</td>
<td>15</td>
<td>12.4</td>
<td>40.4</td>
</tr>
<tr>
<td>MIL</td>
<td>8</td>
<td>3.4</td>
<td>60.2</td>
<td>15.4</td>
<td>20.5</td>
<td>12.2</td>
</tr>
<tr>
<td>OAB</td>
<td>113.2</td>
<td>5.1</td>
<td>28.7</td>
<td>14.6</td>
<td>7.9</td>
<td>3.1</td>
</tr>
<tr>
<td>MTT</td>
<td>166.3</td>
<td>3</td>
<td>3.5</td>
<td>7.3</td>
<td>3</td>
<td>12.7</td>
</tr>
<tr>
<td>IVT</td>
<td>131.3</td>
<td>1.6</td>
<td>2</td>
<td>34.2</td>
<td>2.2</td>
<td>90.7</td>
</tr>
<tr>
<td>DFT</td>
<td>3.2</td>
<td>5.4</td>
<td>28.5</td>
<td>45</td>
<td>9.4</td>
<td>105.8</td>
</tr>
<tr>
<td>SemiB</td>
<td>105.6</td>
<td>327.5</td>
<td>11.9</td>
<td>101.7</td>
<td>24.5</td>
<td>212.2</td>
</tr>
<tr>
<td>CT</td>
<td>11.6</td>
<td>7.2</td>
<td>58.3</td>
<td>8.9</td>
<td>12.8</td>
<td>8.8</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>144.7</td>
<td>5.3</td>
<td>34.4</td>
<td>18.1</td>
<td>3</td>
<td>4.5</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>3.7</td>
<td>6.8</td>
<td>4.2</td>
<td>15.8</td>
<td>3</td>
<td>3.4</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>3.8</td>
<td>5.7</td>
<td>6.1</td>
<td>5.7</td>
<td>2.3</td>
<td>4</td>
</tr>
</tbody>
</table>
Figure 3.3: The survival curves based on the F-score for the entire thirty-three videos and the subset of videos containing different challenging factors, and the average F-scores are shown in the corresponding legend in descending order. (a) Whole set of videos. (b) Occlusion subset of videos. (c) Cluttered background subset of videos. (d) Scale subset of videos. (e) Illumination subset of videos. (f) Pose subset of videos.

**Quantitative Comparison**

The performance of the compared trackers are evaluated quantitatively from two aspects: video-by-video comparison and video set-based comparison. Following [120] [119] [112] [71] [167] [162], three evaluation criteria are used for quantitative comparison: center location error, overlap ratio and success rate. The center location error is defined as the Euclidean distance between the central location of a bounding box and the labeled ground truth. And the overlap ratio is defined as $\frac{\text{area}(B_T \cap B_G)}{\text{area}(B_T \cup B_G)}$, where $B_T$ and $B_G$ are the bounding boxes of the tracker and ground-truth. That a frame is successfully tracked means that the overlap ratio is larger than 0.5. Tables 3.2, 3.3 and 3.4 show the quantitative comparison of evaluated
Table 3.3: VOC Overlapping Rate. The best three results are shown in red, blue and green.

(a)

<table>
<thead>
<tr>
<th></th>
<th>Crossing</th>
<th>Trellis</th>
<th>MountainBike</th>
<th>FaceOcc1</th>
<th>FaceOcc2</th>
<th>Deer</th>
<th>Football</th>
<th>Football1</th>
<th>Skating1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>0.69</td>
<td>0.62</td>
<td>0.71</td>
<td>0.73</td>
<td>0.79</td>
<td>0.75</td>
<td>0.55</td>
<td>0.67</td>
<td>0.31</td>
</tr>
<tr>
<td>LIAPG</td>
<td>0.21</td>
<td>0.2</td>
<td>0.74</td>
<td>0.75</td>
<td>0.69</td>
<td>0.61</td>
<td>0.56</td>
<td>0.57</td>
<td>0.1</td>
</tr>
<tr>
<td>SCM</td>
<td>0.79</td>
<td>0.68</td>
<td>0.68</td>
<td>0.8</td>
<td>0.73</td>
<td>0.75</td>
<td>0.49</td>
<td>0.4</td>
<td>0.47</td>
</tr>
<tr>
<td>CSK</td>
<td>0.49</td>
<td>0.49</td>
<td>0.71</td>
<td>0.8</td>
<td>0.78</td>
<td>0.76</td>
<td>0.56</td>
<td>0.47</td>
<td>0.5</td>
</tr>
<tr>
<td>Frag</td>
<td>0.31</td>
<td>0.29</td>
<td>0.13</td>
<td>0.82</td>
<td>0.65</td>
<td>0.17</td>
<td>0.71</td>
<td>0.37</td>
<td>0.13</td>
</tr>
<tr>
<td>MIL</td>
<td>0.74</td>
<td>0.25</td>
<td>0.46</td>
<td>0.6</td>
<td>0.68</td>
<td>0.13</td>
<td>0.59</td>
<td>0.67</td>
<td>0.13</td>
</tr>
<tr>
<td>OAB</td>
<td>0.67</td>
<td>0.14</td>
<td>0.64</td>
<td>0.66</td>
<td>0.6</td>
<td>0.72</td>
<td>0.34</td>
<td>0.28</td>
<td>0.4</td>
</tr>
<tr>
<td>MTT</td>
<td>0.2</td>
<td>0.22</td>
<td>0.75</td>
<td>0.7</td>
<td>0.75</td>
<td>0.62</td>
<td>0.58</td>
<td>0.56</td>
<td>0.1</td>
</tr>
<tr>
<td>IVT</td>
<td>0.31</td>
<td>0.26</td>
<td>0.73</td>
<td>0.73</td>
<td>0.73</td>
<td>0.03</td>
<td>0.56</td>
<td>0.56</td>
<td>0.07</td>
</tr>
<tr>
<td>DFT</td>
<td>0.5</td>
<td>0.36</td>
<td>0.3</td>
<td>0.77</td>
<td>0.26</td>
<td>0.66</td>
<td>0.87</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>SemiB</td>
<td>0.69</td>
<td>0.2</td>
<td>0.23</td>
<td>0.57</td>
<td>0.48</td>
<td>0.64</td>
<td>0.15</td>
<td>0.13</td>
<td>0.08</td>
</tr>
<tr>
<td>CT</td>
<td>0.69</td>
<td>0.34</td>
<td>0.14</td>
<td>0.64</td>
<td>0.61</td>
<td>0.04</td>
<td>0.61</td>
<td>0.24</td>
<td>0.09</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.77</td>
<td>0.75</td>
<td>0.33</td>
<td>0.67</td>
<td>0.71</td>
<td>0.43</td>
<td>0.59</td>
<td>0.56</td>
<td>0.22</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>0.78</td>
<td>0.8</td>
<td>0.52</td>
<td>0.8</td>
<td>0.73</td>
<td>0.57</td>
<td>0.62</td>
<td>0.77</td>
<td>0.53</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>0.75</td>
<td>0.75</td>
<td>0.68</td>
<td>0.78</td>
<td>0.77</td>
<td>0.73</td>
<td>0.72</td>
<td>0.72</td>
<td>0.73</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th></th>
<th>CarDark</th>
<th>CarScale</th>
<th>Car4</th>
<th>David</th>
<th>David2</th>
<th>David3</th>
<th>Shaking</th>
<th>Basketball</th>
<th>Bolt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>0.9</td>
<td>0.41</td>
<td>0.5</td>
<td>0.24</td>
<td>0.87</td>
<td>0.29</td>
<td>0.35</td>
<td>0.2</td>
<td>0.01</td>
</tr>
<tr>
<td>LIAPG</td>
<td>0.89</td>
<td>0.51</td>
<td>0.25</td>
<td>0.54</td>
<td>0.86</td>
<td>0.38</td>
<td>0.08</td>
<td>0.23</td>
<td>0.01</td>
</tr>
<tr>
<td>SCM</td>
<td>0.85</td>
<td>0.6</td>
<td>0.76</td>
<td>0.73</td>
<td>0.75</td>
<td>0.4</td>
<td>0.69</td>
<td>0.47</td>
<td>0.02</td>
</tr>
<tr>
<td>CSK</td>
<td>0.75</td>
<td>0.42</td>
<td>0.48</td>
<td>0.41</td>
<td>0.81</td>
<td>0.49</td>
<td>0.58</td>
<td>0.71</td>
<td>0.02</td>
</tr>
<tr>
<td>Frag</td>
<td>0.31</td>
<td>0.43</td>
<td>0.19</td>
<td>0.17</td>
<td>0.24</td>
<td>0.68</td>
<td>0.08</td>
<td>0.62</td>
<td>0.14</td>
</tr>
<tr>
<td>MIL</td>
<td>0.2</td>
<td>0.42</td>
<td>0.26</td>
<td>0.43</td>
<td>0.46</td>
<td>0.54</td>
<td>0.43</td>
<td>0.22</td>
<td>0.01</td>
</tr>
<tr>
<td>OAB</td>
<td>0.79</td>
<td>0.4</td>
<td>0.21</td>
<td>0.39</td>
<td>0.32</td>
<td>0.33</td>
<td>0.01</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>MTT</td>
<td>0.83</td>
<td>0.49</td>
<td>0.45</td>
<td>0.3</td>
<td>0.86</td>
<td>0.1</td>
<td>0.04</td>
<td>0.2</td>
<td>0.01</td>
</tr>
<tr>
<td>IVT</td>
<td>0.67</td>
<td>0.63</td>
<td>0.88</td>
<td>0.05</td>
<td>0.71</td>
<td>0.48</td>
<td>0.03</td>
<td>0.11</td>
<td>0.01</td>
</tr>
<tr>
<td>DFT</td>
<td>0.38</td>
<td>0.42</td>
<td>0.24</td>
<td>0.3</td>
<td>0.55</td>
<td>0.56</td>
<td>0.04</td>
<td>0.61</td>
<td>0.03</td>
</tr>
<tr>
<td>SemiB</td>
<td>0.84</td>
<td>0.33</td>
<td>0.23</td>
<td>0.25</td>
<td>0.55</td>
<td>0.15</td>
<td>0.01</td>
<td>0.02</td>
<td>0.06</td>
</tr>
<tr>
<td>CT</td>
<td>0</td>
<td>0.44</td>
<td>0.21</td>
<td>0.5</td>
<td>0</td>
<td>0.31</td>
<td>0.1</td>
<td>0.26</td>
<td>0.01</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.75</td>
<td>0.52</td>
<td>0.71</td>
<td>0.57</td>
<td>0.79</td>
<td>0.56</td>
<td>0.09</td>
<td>0.03</td>
<td>0.02</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>0.7</td>
<td>0.7</td>
<td>0.86</td>
<td>0.62</td>
<td>0.74</td>
<td>0.7</td>
<td>0.35</td>
<td>0.09</td>
<td>0.02</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>0.74</td>
<td>0.66</td>
<td>0.86</td>
<td>0.66</td>
<td>0.85</td>
<td>0.7</td>
<td>0.73</td>
<td>0.58</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>Dog1</td>
<td>Dudek</td>
<td>Fish</td>
<td>FleetFace</td>
<td>Freeman1</td>
<td>Freeman2</td>
<td>Jumping</td>
<td>Singer1</td>
<td>Singer2</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>Struck</td>
<td>0.55</td>
<td>0.73</td>
<td>0.86</td>
<td>0.61</td>
<td>0.34</td>
<td>0.27</td>
<td>0.62</td>
<td>0.37</td>
<td>0.04</td>
</tr>
<tr>
<td>L1APG</td>
<td>0.71</td>
<td>0.69</td>
<td>0.35</td>
<td>0.46</td>
<td>0.21</td>
<td>0.36</td>
<td>0.15</td>
<td>0.29</td>
<td>0.04</td>
</tr>
<tr>
<td>SCM</td>
<td>0.71</td>
<td>0.77</td>
<td>0.75</td>
<td>0.61</td>
<td>0.62</td>
<td>0.73</td>
<td>0.12</td>
<td>0.87</td>
<td>0.17</td>
</tr>
<tr>
<td>CSK</td>
<td>0.55</td>
<td>0.72</td>
<td>0.21</td>
<td>0.59</td>
<td>0.24</td>
<td>0.3</td>
<td>0.05</td>
<td>0.36</td>
<td>0.04</td>
</tr>
<tr>
<td>Frag</td>
<td>0.55</td>
<td>0.54</td>
<td>0.55</td>
<td>0.47</td>
<td>0.38</td>
<td>0.33</td>
<td>0.68</td>
<td>0.21</td>
<td>0.2</td>
</tr>
<tr>
<td>MIL</td>
<td>0.54</td>
<td>0.71</td>
<td>0.46</td>
<td>0.49</td>
<td>0.35</td>
<td>0.01</td>
<td>0.53</td>
<td>0.36</td>
<td>0.52</td>
</tr>
<tr>
<td>OAB</td>
<td>0.55</td>
<td>0.66</td>
<td>0.04</td>
<td>0.53</td>
<td>0.36</td>
<td>0.27</td>
<td>0.07</td>
<td>0.34</td>
<td>0.05</td>
</tr>
<tr>
<td>MTT</td>
<td>0.69</td>
<td>0.76</td>
<td>0.17</td>
<td>0.51</td>
<td>0.22</td>
<td>0.47</td>
<td>0.3</td>
<td>0.34</td>
<td>0.04</td>
</tr>
<tr>
<td>IVT</td>
<td>0.74</td>
<td>0.75</td>
<td>0.77</td>
<td>0.46</td>
<td>0.43</td>
<td>0.41</td>
<td>0.12</td>
<td>0.58</td>
<td>0.04</td>
</tr>
<tr>
<td>DFT</td>
<td>0.44</td>
<td>0.69</td>
<td>0.76</td>
<td>0.48</td>
<td>0.4</td>
<td>0.32</td>
<td>0.11</td>
<td>0.36</td>
<td>0.63</td>
</tr>
<tr>
<td>SemiB</td>
<td>0.45</td>
<td>0.38</td>
<td>0.21</td>
<td>0.26</td>
<td>0.13</td>
<td>0.15</td>
<td>0.06</td>
<td>0.12</td>
<td>0.03</td>
</tr>
<tr>
<td>CT</td>
<td>0.54</td>
<td>0.65</td>
<td>0.72</td>
<td>0.52</td>
<td>0.15</td>
<td>0</td>
<td>0.05</td>
<td>0.35</td>
<td>0.08</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.66</td>
<td>0.7</td>
<td>0.73</td>
<td>0.52</td>
<td>0.4</td>
<td>0.65</td>
<td>0.67</td>
<td>0.7</td>
<td>0.32</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>0.78</td>
<td>0.83</td>
<td>0.77</td>
<td>0.7</td>
<td>0.67</td>
<td>0.74</td>
<td>0.6</td>
<td>0.79</td>
<td>0.57</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>0.75</td>
<td>0.82</td>
<td>0.84</td>
<td>0.71</td>
<td>0.6</td>
<td>0.74</td>
<td>0.72</td>
<td>0.77</td>
<td>0.65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Subway</th>
<th>Walking</th>
<th>Walking2</th>
<th>Sylvester</th>
<th>Mhyang</th>
<th>Boy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>0.66</td>
<td>0.59</td>
<td>0.53</td>
<td>0.73</td>
<td>0.82</td>
<td>0.76</td>
</tr>
<tr>
<td>L1APG</td>
<td>0.16</td>
<td>0.76</td>
<td>0.76</td>
<td>0.41</td>
<td>0.83</td>
<td>0.74</td>
</tr>
<tr>
<td>SCM</td>
<td>0.73</td>
<td>0.72</td>
<td>0.82</td>
<td>0.69</td>
<td>0.81</td>
<td>0.38</td>
</tr>
<tr>
<td>CSK</td>
<td>0.2</td>
<td>0.55</td>
<td>0.47</td>
<td>0.64</td>
<td>0.8</td>
<td>0.66</td>
</tr>
<tr>
<td>Frag</td>
<td>0.47</td>
<td>0.55</td>
<td>0.28</td>
<td>0.59</td>
<td>0.65</td>
<td>0.39</td>
</tr>
<tr>
<td>MIL</td>
<td>0.66</td>
<td>0.55</td>
<td>0.29</td>
<td>0.53</td>
<td>0.51</td>
<td>0.5</td>
</tr>
<tr>
<td>OAB</td>
<td>0.17</td>
<td>0.55</td>
<td>0.37</td>
<td>0.56</td>
<td>0.75</td>
<td>0.8</td>
</tr>
<tr>
<td>MTT</td>
<td>0.07</td>
<td>0.67</td>
<td>0.79</td>
<td>0.65</td>
<td>0.85</td>
<td>0.51</td>
</tr>
<tr>
<td>IVT</td>
<td>0.17</td>
<td>0.77</td>
<td>0.8</td>
<td>0.52</td>
<td>0.8</td>
<td>0.26</td>
</tr>
<tr>
<td>DFT</td>
<td>0.74</td>
<td>0.57</td>
<td>0.41</td>
<td>0.38</td>
<td>0.71</td>
<td>0.4</td>
</tr>
<tr>
<td>SemiB</td>
<td>0.29</td>
<td>0.2</td>
<td>0.47</td>
<td>0.34</td>
<td>0.67</td>
<td>0.32</td>
</tr>
<tr>
<td>CT</td>
<td>0.58</td>
<td>0.53</td>
<td>0.27</td>
<td>0.67</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.19</td>
<td>0.59</td>
<td>0.35</td>
<td>0.64</td>
<td>0.81</td>
<td>0.7</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>0.68</td>
<td>0.45</td>
<td>0.7</td>
<td>0.63</td>
<td>0.81</td>
<td>0.76</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>0.7</td>
<td>0.51</td>
<td>0.59</td>
<td>0.69</td>
<td>0.76</td>
<td>0.71</td>
</tr>
</tbody>
</table>
Table 3.4: Success Rate. The best three results are shown in red, blue and green.

(a)

<table>
<thead>
<tr>
<th></th>
<th>Crossing</th>
<th>Trellis</th>
<th>MountainBike</th>
<th>FaceOcc1</th>
<th>FaceOcc2</th>
<th>Deer</th>
<th>Football</th>
<th>Football1</th>
<th>Skating1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>0.96</td>
<td>0.78</td>
<td>0.86</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.67</td>
<td>0.88</td>
<td>0.37</td>
</tr>
<tr>
<td>L1APG</td>
<td>0.25</td>
<td>0.16</td>
<td>0.93</td>
<td>1</td>
<td>0.81</td>
<td>0.72</td>
<td>0.69</td>
<td>0.59</td>
<td>0.13</td>
</tr>
<tr>
<td>SCM</td>
<td>1</td>
<td>0.85</td>
<td>0.96</td>
<td>1</td>
<td>0.88</td>
<td>0.03</td>
<td>0.6</td>
<td>0.41</td>
<td>0.44</td>
</tr>
<tr>
<td>CSK</td>
<td>0.33</td>
<td>0.64</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.66</td>
<td>0.39</td>
<td>0.37</td>
</tr>
<tr>
<td>Frag</td>
<td>0.38</td>
<td>0.36</td>
<td>0.14</td>
<td>1</td>
<td>0.76</td>
<td>0.21</td>
<td>0.92</td>
<td>0.34</td>
<td>0.13</td>
</tr>
<tr>
<td>MIL</td>
<td>0.99</td>
<td>0.24</td>
<td>0.58</td>
<td>0.77</td>
<td>0.94</td>
<td>0.13</td>
<td>0.74</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>OAB</td>
<td>0.84</td>
<td>0.18</td>
<td>0.92</td>
<td>0.91</td>
<td>0.76</td>
<td>0.96</td>
<td>0.36</td>
<td>0.28</td>
<td>0.34</td>
</tr>
<tr>
<td>MTT</td>
<td>0.23</td>
<td>0.2</td>
<td>0.97</td>
<td>1</td>
<td>0.9</td>
<td>0.87</td>
<td>0.78</td>
<td>0.59</td>
<td>0.14</td>
</tr>
<tr>
<td>IVT</td>
<td>0.24</td>
<td>0.32</td>
<td>0.99</td>
<td>0.98</td>
<td>0.92</td>
<td>0.03</td>
<td>0.72</td>
<td>0.64</td>
<td>0.08</td>
</tr>
<tr>
<td>DFT</td>
<td>0.66</td>
<td>0.52</td>
<td>0.35</td>
<td>0.81</td>
<td>1</td>
<td>0.31</td>
<td>0.84</td>
<td>1</td>
<td>0.16</td>
</tr>
<tr>
<td>SemiB</td>
<td>0.88</td>
<td>0.2</td>
<td>0.29</td>
<td>0.71</td>
<td>0.56</td>
<td>0.86</td>
<td>0.18</td>
<td>0.14</td>
<td>0.08</td>
</tr>
<tr>
<td>CT</td>
<td>0.99</td>
<td>0.36</td>
<td>0.17</td>
<td>0.86</td>
<td>0.76</td>
<td>0.04</td>
<td>0.78</td>
<td>0.08</td>
<td>0.11</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>1</td>
<td>0.91</td>
<td>0.4</td>
<td>0.85</td>
<td>0.86</td>
<td>0.58</td>
<td>0.76</td>
<td>0.72</td>
<td>0.23</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>1</td>
<td>0.98</td>
<td>0.68</td>
<td>1</td>
<td>0.95</td>
<td>0.79</td>
<td>0.79</td>
<td>0.86</td>
<td>0.61</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>1</td>
<td>0.94</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.87</td>
<td>0.93</td>
<td>0.99</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th></th>
<th>CarDark</th>
<th>CarScale</th>
<th>Car4</th>
<th>David</th>
<th>David2</th>
<th>David3</th>
<th>Shaking</th>
<th>Basketball</th>
<th>Bolt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>1</td>
<td>0.43</td>
<td>0.41</td>
<td>0.24</td>
<td>1</td>
<td>0.34</td>
<td>0.17</td>
<td>0.1</td>
<td>0.02</td>
</tr>
<tr>
<td>L1APG</td>
<td>1</td>
<td>0.59</td>
<td>0.3</td>
<td>0.7</td>
<td>1</td>
<td>0.46</td>
<td>0.04</td>
<td>0.29</td>
<td>0.01</td>
</tr>
<tr>
<td>SCM</td>
<td>1</td>
<td>0.65</td>
<td>0.97</td>
<td>0.92</td>
<td>0.93</td>
<td>0.48</td>
<td>0.9</td>
<td>0.62</td>
<td>0.01</td>
</tr>
<tr>
<td>CSK</td>
<td>0.99</td>
<td>0.46</td>
<td>0.28</td>
<td>0.42</td>
<td>1</td>
<td>0.63</td>
<td>0.58</td>
<td>0.88</td>
<td>0.02</td>
</tr>
<tr>
<td>Frag</td>
<td>0.26</td>
<td>0.44</td>
<td>0.21</td>
<td>0.13</td>
<td>0.3</td>
<td>0.82</td>
<td>0.07</td>
<td>0.71</td>
<td>0.11</td>
</tr>
<tr>
<td>MIL</td>
<td>0.18</td>
<td>0.46</td>
<td>0.28</td>
<td>0.25</td>
<td>0.33</td>
<td>0.69</td>
<td>0.23</td>
<td>0.28</td>
<td>0.01</td>
</tr>
<tr>
<td>OAB</td>
<td>0.95</td>
<td>0.45</td>
<td>0.28</td>
<td>0.15</td>
<td>0.25</td>
<td>0.34</td>
<td>0.01</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>MTT</td>
<td>1</td>
<td>0.57</td>
<td>0.32</td>
<td>0.29</td>
<td>1</td>
<td>0.1</td>
<td>0.01</td>
<td>0.29</td>
<td>0.01</td>
</tr>
<tr>
<td>IVT</td>
<td>0.7</td>
<td>0.7</td>
<td>1</td>
<td>0.8</td>
<td>0.96</td>
<td>0.64</td>
<td>0.01</td>
<td>0.1</td>
<td>0.01</td>
</tr>
<tr>
<td>DFT</td>
<td>0.34</td>
<td>0.46</td>
<td>0.26</td>
<td>0.24</td>
<td>0.54</td>
<td>0.75</td>
<td>0.82</td>
<td>0.72</td>
<td>0.04</td>
</tr>
<tr>
<td>SemiB</td>
<td>0.93</td>
<td>0.4</td>
<td>0.25</td>
<td>0.21</td>
<td>0.59</td>
<td>0.18</td>
<td>0.01</td>
<td>0.02</td>
<td>0.07</td>
</tr>
<tr>
<td>CT</td>
<td>0</td>
<td>0.46</td>
<td>0.28</td>
<td>0.44</td>
<td>0</td>
<td>0.35</td>
<td>0.04</td>
<td>0.27</td>
<td>0.01</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.94</td>
<td>0.66</td>
<td>1</td>
<td>0.7</td>
<td>0.95</td>
<td>0.74</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>0.98</td>
<td>0.79</td>
<td>1</td>
<td>0.8</td>
<td>0.95</td>
<td>0.92</td>
<td>0.35</td>
<td>0.11</td>
<td>0.03</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>1</td>
<td>0.71</td>
<td>1</td>
<td>0.89</td>
<td>1</td>
<td>0.96</td>
<td>0.99</td>
<td>0.75</td>
<td>0.0</td>
</tr>
</tbody>
</table>
(c)

<table>
<thead>
<tr>
<th></th>
<th>Dog1</th>
<th>Dudek</th>
<th>Fish</th>
<th>FleetFace</th>
<th>Freeman1</th>
<th>Freeman3</th>
<th>Jumping</th>
<th>Singer1</th>
<th>Singer2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>0.65</td>
<td><strong>0.98</strong></td>
<td>1</td>
<td>0.67</td>
<td>0.22</td>
<td>0.21</td>
<td>0.81</td>
<td>0.31</td>
<td>0.04</td>
</tr>
<tr>
<td>L1APG</td>
<td>0.85</td>
<td>0.8</td>
<td>0.04</td>
<td>0.55</td>
<td>0.15</td>
<td>0.36</td>
<td>0.12</td>
<td>0.38</td>
<td>0.04</td>
</tr>
<tr>
<td>SCM</td>
<td>0.85</td>
<td><strong>0.98</strong></td>
<td>0.86</td>
<td><strong>0.71</strong></td>
<td><strong>0.81</strong></td>
<td><strong>0.94</strong></td>
<td>0.12</td>
<td>1</td>
<td>0.16</td>
</tr>
<tr>
<td>CSK</td>
<td>0.65</td>
<td>0.95</td>
<td>0.04</td>
<td>0.68</td>
<td>0.14</td>
<td>0.33</td>
<td>0.05</td>
<td>0.3</td>
<td>0.04</td>
</tr>
<tr>
<td>Frag</td>
<td>0.62</td>
<td>0.59</td>
<td>0.55</td>
<td>0.46</td>
<td>0.2</td>
<td>0.33</td>
<td><strong>0.86</strong></td>
<td>0.22</td>
<td>0.2</td>
</tr>
<tr>
<td>MIL</td>
<td>0.65</td>
<td>0.86</td>
<td>0.39</td>
<td>0.54</td>
<td>0.15</td>
<td>0.01</td>
<td>0.48</td>
<td>0.28</td>
<td>0.49</td>
</tr>
<tr>
<td>OAB</td>
<td>0.65</td>
<td>0.8</td>
<td>0.04</td>
<td>0.68</td>
<td>0.26</td>
<td>0.2</td>
<td>0.05</td>
<td>0.24</td>
<td>0.03</td>
</tr>
<tr>
<td>MTT</td>
<td>0.79</td>
<td>0.93</td>
<td>0.04</td>
<td>0.55</td>
<td>0.17</td>
<td>0.48</td>
<td>0.09</td>
<td>0.34</td>
<td>0.04</td>
</tr>
<tr>
<td>IVT</td>
<td><strong>0.87</strong></td>
<td>0.97</td>
<td>1</td>
<td>0.47</td>
<td>0.33</td>
<td>0.46</td>
<td>0.1</td>
<td>0.52</td>
<td>0.04</td>
</tr>
<tr>
<td>DFT</td>
<td>0.52</td>
<td>0.8</td>
<td>0.86</td>
<td>0.56</td>
<td>0.19</td>
<td>0.35</td>
<td>0.12</td>
<td>0.28</td>
<td><strong>0.7</strong></td>
</tr>
<tr>
<td>SemiB</td>
<td>0.55</td>
<td>0.46</td>
<td>0.12</td>
<td>0.34</td>
<td>0.16</td>
<td>0.19</td>
<td>0.05</td>
<td>0.15</td>
<td>0.03</td>
</tr>
<tr>
<td>CT</td>
<td>0.65</td>
<td>0.86</td>
<td>0.89</td>
<td>0.64</td>
<td>0.1</td>
<td>0</td>
<td>0.01</td>
<td>0.25</td>
<td>0.01</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.76</td>
<td>0.97</td>
<td>0.84</td>
<td>0.68</td>
<td>0.22</td>
<td>0.78</td>
<td><strong>0.88</strong></td>
<td>0.7</td>
<td>0.39</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td><strong>0.96</strong></td>
<td>1</td>
<td>1</td>
<td><strong>0.9</strong></td>
<td><strong>0.92</strong></td>
<td>1</td>
<td>0.79</td>
<td><strong>0.99</strong></td>
<td>0.65</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>0.86</td>
<td>1</td>
<td>1</td>
<td><strong>0.99</strong></td>
<td>0.79</td>
<td>1</td>
<td>0.96</td>
<td>1</td>
<td><strong>0.92</strong></td>
</tr>
</tbody>
</table>

(d)

<table>
<thead>
<tr>
<th></th>
<th>Subway</th>
<th>Walking</th>
<th>Walking2</th>
<th>Sylvester</th>
<th>Mhyang</th>
<th>Boy</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Struck</td>
<td>0.94</td>
<td>0.64</td>
<td>0.46</td>
<td><strong>0.93</strong></td>
<td>1</td>
<td><strong>0.98</strong></td>
<td>18</td>
</tr>
<tr>
<td>L1APG</td>
<td>0.23</td>
<td>1</td>
<td>0.98</td>
<td>0.43</td>
<td>0.98</td>
<td>0.9</td>
<td>22</td>
</tr>
<tr>
<td>SCM</td>
<td>1</td>
<td>0.96</td>
<td>1</td>
<td><strong>0.89</strong></td>
<td>1</td>
<td>0.44</td>
<td>1</td>
</tr>
<tr>
<td>CSK</td>
<td>0.22</td>
<td>0.55</td>
<td>0.4</td>
<td>0.74</td>
<td>1</td>
<td>0.84</td>
<td>360</td>
</tr>
<tr>
<td>Frag</td>
<td>0.59</td>
<td>0.52</td>
<td>0.35</td>
<td>0.69</td>
<td>0.73</td>
<td>0.46</td>
<td>5</td>
</tr>
<tr>
<td>MIL</td>
<td>0.81</td>
<td>0.55</td>
<td>0.39</td>
<td>0.55</td>
<td>0.39</td>
<td>0.41</td>
<td>36</td>
</tr>
<tr>
<td>OAB</td>
<td>0.22</td>
<td>0.49</td>
<td>0.4</td>
<td>0.68</td>
<td>0.96</td>
<td><strong>0.99</strong></td>
<td>20</td>
</tr>
<tr>
<td>MTT</td>
<td>0.08</td>
<td><strong>0.97</strong></td>
<td><strong>0.99</strong></td>
<td>0.83</td>
<td>1</td>
<td>0.49</td>
<td>3</td>
</tr>
<tr>
<td>IVT</td>
<td>0.21</td>
<td>1</td>
<td>1</td>
<td>0.68</td>
<td>1</td>
<td>0.33</td>
<td>30</td>
</tr>
<tr>
<td>DFT</td>
<td><strong>0.99</strong></td>
<td>0.56</td>
<td>0.39</td>
<td>0.41</td>
<td>0.78</td>
<td>0.48</td>
<td>11</td>
</tr>
<tr>
<td>SemiB</td>
<td>0.38</td>
<td>0.25</td>
<td>0.38</td>
<td>0.43</td>
<td>0.77</td>
<td>0.38</td>
<td>11</td>
</tr>
<tr>
<td>CT</td>
<td>0.8</td>
<td>0.53</td>
<td>0.39</td>
<td>0.83</td>
<td>0.75</td>
<td>0.7</td>
<td>78</td>
</tr>
<tr>
<td>JSRFFT</td>
<td>0.23</td>
<td>0.69</td>
<td>0.39</td>
<td>0.83</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
</tr>
<tr>
<td>RJSRFFT</td>
<td>0.94</td>
<td>0.33</td>
<td>0.93</td>
<td>0.8</td>
<td>1</td>
<td><strong>0.99</strong></td>
<td>0.6</td>
</tr>
<tr>
<td>K-RJSRFFT</td>
<td>0.98</td>
<td>0.51</td>
<td>0.59</td>
<td><strong>0.96</strong></td>
<td>0.98</td>
<td>0.95</td>
<td>0.5</td>
</tr>
</tbody>
</table>
trackers in terms of center location error, overlap ratio and success rate, respectively. And the last column of Table 3.4(d) shows the average frame-per-second running time of the compared trackers. The proposed trackers RJSRFFT and K-RJSRFFT rank in the top three among all trackers. The proposed trackers RJSRFFT and K-RJSRFFT rank in the top three among all trackers. The proposed trackers achieve much better results compared with sparse representation based tracker using single feature, i.e., L1APG, SCM and MTT. This is because our tracker is based on multi-feature sparse representations which can dynamically utilize different features to handle different kinds of variations while preserving the advantages of sparse representation. The proposed multi-cue trackers also perform better than existing multi-cue trackers, i.e., OAB, MIL, and SemiB. This is because the more informative features are dynamically exploited for fusion at feature level than that of score level. But feature level fusion-based tracker JSRFFT does not perform as well as R-JSRFFT and K-RJSRFFT, which suggests that unreliable feature detection scheme is essential for feature-level fusion. Comparing the performance of RJSRFFT and JSRFFT suggests that unreliable features detection can improve the performance of joint sparse representation based tracker. K-RJSRFFT outperforms RJSRFFT which indicates that exploiting the non-linearity of features can improve the performance of multi-cue tracker.

Recently, Smeulders et al. presented a comprehensive survey and systematic evaluation on state-of-the-art visual tracking algorithms in [100] based on the Amsterdam Library of Ordinary Videos data set, named ALOV++. The evaluation protocol in [100] shows that the F-score is an effective metric to evaluate a tracker’s accuracy, and the survival curve as illustrated in [100] provides a cumulative rendition of the quality of the tracker on a set of videos, which avoids the risk of being trapped in the peculiarity of the single video instance. Therefore, to conduct the video set-based comparison, we compute the F-score using the tracking results of each tracker on each video, and plot the corresponding survival curve on the entire thirty-three videos and the subset of videos containing different
challenging factors as shown in Table 3.1. The F-score for each video is defined as $2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$, where $\text{precision} = \frac{n_{tp}}{n_{tp} + n_{fp}}$, $\text{recall} = \frac{n_{tp}}{n_{tp} + n_{fn}}$, and $n_{tp}, n_{fp}, n_{fn}$ denote the number of true positives, false positives, false negatives in a video. After getting the F-score of each video in the video set, the sets of videos are sorted in the descending order and the curve survival curve of F-score is plotted with respect to the sorted videos. Fig. 3.3 illustrates the survival curves based on F-score on the whole thirty-three videos and the subset of videos containing different challenging factors. We can see that the K-RJSRFFT method achieve the best performance on the whole video and subsets of video containing different factors in term of average F-core. By performing feature-level fusion in multiple kernel space, the K-RJSRFFT method perform better than the JSRFFT method. With the unreliable feature detection scheme, both K-RJSRFFT and RJSRFFT method perform better than the JSRFFT method, which shows the effectiveness of the unreliable feature detection scheme. By utilizing the complementarity of multiple reliable features, both K-RJSRFFT and RJSRFFT method show superior performance under different challenging factors, especially in occlusion, illumination and cluttered background.

**Qualitative Comparison**

We qualitatively evaluate the trackers in five aspects based on some typical video sequences as follows:

**Occlusion:** For the video *Occluded Face 1*, a woman face undergoes some partial and severe occlusion by a book in some frames. Except that the CT, DFT, JSRFFT and MIL methods have small drift when the woman’s face is severely occluded around Frame 730 and Frame 873, all other trackers can successfully track the woman’s face through the whole video. For the video *David Outdoor*, a man is walking with body deformation and occlusion. The contrast between the target and background is low which makes the SemiB, MIL, CT, L1APG, Struck, JSRFFT methods lose the target around Frame 49, 108, 120, 195, 238, respectively. The full
Figure 3.4: Qualitative results on some typical frames including some challenging factors. (a) Occlusion. (b) Background. (c) Scale. (d) Illumination. (d) Pose.
body occlusion also cause the tracking drift problem of the MTT, OAB, SCM, C-SK, JSRFFT, DFT, IVT methods around Frame 27, 83, 188, respectively. Because the K-RJSRFFT method combines local and global features from non-linear kernel space to account for the variation in foreground and background, it performs well in the whole sequence. For the sequence Walking2, a woman is walking with change of scale. Most trackers perform well except that the Frag, OAB, SemiB, CT, MIL, JSRFFT methods drift away form the woman due to the occlusion caused by a man with similar appearance. As can be seen from above comparison, the proposed trackers RJSRFFT and K-RJSRFFT can handle occlusion well. With the similar merit of [49], the fusion of local features enables it to be less sensitive to partial occlusion. Besides, different from the JSRFFT method which directly performs fusion of all features, unreliable feature detection scheme in the RJSRFFT and K-RJSRFFT methods can improve the tracking performance of joint sparse representation based trackers since the features extracted from occluded object may not be reliable.

**Cluttered Background:** In the video Crossing, the contrast between the walking person and the background is low, which leads to the drift problem of some trackers, i.e., the MTT, L1APG, DFT and Frag methods around Frame 55 and 83. In the video Mountain-bike, a man riding a mountain bike undergoes pose variation with cluttered background. The cluttered background makes the CT, Frag, DFT, and JSRFFT methods drift away from the tracked object around Frame 50 and 95, respectively. In the video Jumping, A man is jumping rope with motion blur occurring on his face. The cluttered background makes it easy for trackers such as the IVT, SCM, L1APG, MTT, OAM, CT, DFT, CSK, SemiB methods drift from the blurred face as shown in Frame 171.

The proposed tracker K-RJSRFFT performs well under a cluttered background, and shows improved results compared with the RJSRFFT method. This is because different from the RJSRFFT method, the K-RJSRFFT method utilizes the non-linear similarity of feature and performs feature fusion in a more discriminative kernel space, which enhances the discriminability of appearance model and makes
it less sensitive to cluttered background.

**Scale:** In the video *Car4*, there is a drastic change of scale and illumination when the car goes underneath the overpass. Only the CSK, SCM, IVT, RJSRFFT, K-RJSRFFT, Struck and JSRFFT methods can perform well in the whole sequence while the other trackers have some small or large drift away from the car. In the *David indoor* sequence, a man is walking out of the dark room with a gradual change in scale and pose. The IVT, SCM, MIL, RJSRFFT, K-RJSRFFT, JSRFFT methods perform well on this sequence. Other trackers have some drift when the man undergoes pose and scale variation. (e.g. Frame 166). In the *Freeman3* sequence, a man is walking in the classroom from the back to the front with a gradual change in scale. Due to the similar appearance of human face in the background and the scale change of the target, the OAB, Struck, Frag, CT, MIL, CSK, IVT, L1APG, DFT methods lose the tracking of the target. The K-RJSRFFT, RJSRFFT and JSRFFT methods perform well on this sequence.

**Illumination:** For the challenging video *Trellis*, the object appearance changes significantly because of illumination and pose. the OAB, SemiB, IVT, MTT, L1APG, CT, MIL methods lose the target due to the cast shadow on the face. When the object changes its pose, the SCM method drifts away from the target. Only the R-JSRFFT, K-RJSRFFT, and JSRFFT methods perform well in the whole sequence as a result of the fusion of local illumination insensitive feature which enables them to handle such kinds of variations. Video *Skating1* is also challenging in which a woman is skating under variations of pose, illumination and background. Some partial occlusion also occurs as shown in Frame 174. Except the proposed trackers K-RJSRFFT and RJSRFFT, other trackers do not perform well in the whole sequence. And the dramatic change of lighting in the background let the RJSRFFT method drift away from target while the K-RJSRFFT method still tracks the target stably which shows that fusion features in kernel space can enhance the discriminability of feature fusion based trackers. In the sequence *Car11*, the significant change of illumination and low resolution of object makes tracking difficult. Except the DFT,
CT, Frag, MIL methods, all others trackers are able to track the target with high success rate.

**Pose:** The *Shaking* sequence is challenging because of significant pose variations. The proposed method K-RJSRFFT can successfully track the target. The RJSRFFT, OAB, MIL, CSK methods have small drift away from the target due to the pose variation. Other trackers such as the JSRFFT, CT, IVT, SemiB, MTT, L1APG methods fail to track the target due to the combined variation of pose and occlusion as well as illumination around Frame 59. For the *Basketball* sequence, the target is a basketball player running on the court who undergoes significant pose variation as well as some partial occlusion. The Struck, OAB, JSRFFT methods drift from the target around Frame 64. And when another player with similar appearance goes close to the player around Frame 302, the MIL, CT and L1APG method lose the target. Only the K-RJSRFFT, DFT, CSK and Frag methods perform well in this sequence. In the sequence *Sylvester*, the object undergoes significant pose variation around frame 1091, and the DFT, IVT, SCM, L1APG, Frag and MIL method drift away from the target. Our proposed trackers K-RJSRFFT and RJSRFFT are able to track the target with high success rate.

### 3.6 Conclusion

In this chapter, we have formulated a feature-level fusion visual tracker based on joint sparse representation. The proposed method outperforms other feature fusion-based trackers and sparse representation-based trackers under appearance variations such as occlusion, scale, illumination and poses which can be shown in the experimental results. This demonstrates that the proposed method for robust feature-level fusion is effective.
Chapter 4

Multiple Sparse Representations with Commonality and Diversity Modeling for Multi-Cue Visual Tracking

4.1 Introduction

With multiple features of object appearance, a key problem is how to learn a fused representation from multiple visual cues for appearance modeling. Motivated by the recent success of sparse representation in pattern classification [125], several efforts have also been made to develop multi-cue/feature trackers based on sparse representation and show promising results. With the assumption that different visual cues of the tracked object are strongly correlated, instead of learning sparse representation for each visual cue independently, most existing multi-cue sparse trackers attempt to learn the common or similar representations collaboratively. For example, Wu et al.[127] concatenated multiple sources of data and learned the sparse pattern for the concatenated feature using $\ell_1$ norm minimization, which exploits the commonalities of multiple visual cues by implicitly enforcing all the sparse representations of dif-
ferent visual cues to be the same. To learn a robust fused representation of multiple visual cues, Hu et al. [48] employed a multi-task joint sparse representation [142] with a weighted $\ell_{1,2}$ regularization. Although their method does not enforce the representations of all visual cues to be the same, it still requires that the representations have the same positions of non-zero sparse coefficients. Such $\ell_{1,2}$ regularization scheme is also exploited in the multi-task multi-view tracker [44], which still strictly restrict non-outlier tasks to share the same set of features from multiple views.

Generally speaking, employing the commonalities among their representations is reasonable, as different features are extracted from the same object and hence should share some similarity in their underlying representations. However, as each visual cue also describes different features of the tracked object (e.g. edge, color, texture), their own distinctive properties which would complement each other in appearance modeling should also be considered. As observed from Fig. 4.1 which illustrates the representations of different features for a subject’s face under illumination and pose variation, both commonalities and diversities indeed exist in the sparse representations of different visual cues. Actually, as discussed in [166], commonality is closely related to the concepts of agreement or consensus, while complementarity is related to disagreement or diversity. The benefit of simultaneously employing such commonality and diversity have been well demonstrated in feature fusion for many pattern classification tasks [137, 79, 47] because of the fully utilization of shared and complementary information from multiple features. Therefore, it is important to allow these features to have some diversities to reflect their distinctive properties in their representations when employing their commonality, which can further unleash their complementarity in appearance modeling.

In addition, for multi-cue sparse trackers, the commonality among multiple sparse representations determines the common set of object templates from different features that are fused for appearance modeling. As discussed in [48], directly imposing the sparsity constraint onto the commonality may not be able to select object templates with similar appearance in multiple cues for object representation. That
is to say, the commonalities learned under sparsity constraint may not be accurate shared patterns of the tracked object. If non-representative and dissimilar templates with multiple visual cues are selected for appearance modeling, it is more likely that the background image patch is decided as the final result [48]. Although Hu et al. [48] incorporated proximity information based on Euclidean distance to favor the activation of more similar and representative templates for appearance modeling, the metric based on the predefined Euclidean distance may not be able to achieve the closest matching between the object and the templates [56]. As such, the metric which measures the appearance proximity between the tracked object and the templates should also be adaptively updated during the tracking process.

To overcome the aforementioned limitations of existing multi-cue sparsity-based trackers in commonality and diversity modeling, the chapter proposes a novel multiple sparse representation framework for visual tracking using multiple features. Compared to existing trackers based on multiple sparse representations that only consider the commonalities among multiple sparse representations, e.g. [48, 64], the proposed method aims to model the commonalities and diversities among multiple sparse representations by using sparsity pattern decomposition within a unified framework, leading to a more flexible and informative appearance model using multiple visual cues. In addition, we introduce a novel online multiple metric learning algorithm to adaptively and efficiently incorporate the proximity constraint into the proposed framework, so that templates with similar appearance in multiple visual cues can be jointly selected for sparse representation, which ensures the learned commonalities among multiple visual cues is more accurate in appearance modeling.

The rest of this chapter is organized as follows. In Section 4.2, we review some recent works which utilize commonality and diversity of multiple features/modalities for other pattern classification and recognition tasks. In Section 4.3, we present the proposed tracking framework as well as the corresponding optimization algorithm. We further discuss the implementation details in Section 4.4. Experiment results
Figure 4.1: Graphical Illustration of Commonality and Diversity of Feature Representation. (a) A man whose face undergoes illumination and pose variation. (b) The sparse pattern for each visual cue including the HOG and the local covariance descriptors obtained by solving an $\ell_1$ norm minimization problem. (c) The diverse pattern. (d) The common pattern. The $y$-coordinate of all above figures denotes the value of the sparse patterns, and the $x$-coordinate denotes the index of the template in the template set.
and summary are given in Sections 4.5 and 4.6, respectively.

4.2 Related Work

In this section, we review some recent works in commonality and diversity modeling for other pattern classification and recognition tasks for better understanding of the importance of simultaneous utilization of the commonality and the diversity.

4.2.1 Commonality and Diversity Modeling in Pattern Classification and Recognition

Explicitly modeling the commonality and diversity in multiple features/modalities, which is able to simultaneously exploit the shared and the feature/modality-specific information among multiple features and modalities, has been shown to be beneficial for classification and recognition in recent research studies. Liu et al. [79] proposed a partially shared latent factor learning method with a novel NMF model to learn the shared and the diverse latent factors of multiple view data, which exploits the commonality and the diversity of multiple views for multiple-view semi-supervised learning and achieves better result than the methods which exploit the commonality only. To exploit the similarity and the distinctiveness of different features, Yang et al. [137] proposed a novel collaborative representation model which jointly considers the similarity and the diversities of the feature patterns in the feature coding stage. To achieve a more robust fusion of heterogeneous features for RGB-D activity recognition, Hu et al. [47] proposed to exploit the shared information and some diverse specific structures of features from different channels for heterogeneous feature learning. In [109], a multi-modal sharable and specific feature learning algorithm is proposed to obtain the features which reflect their shared properties as well as the modal-specific properties for RGB-D object recognition.
4.3 Proposed Tracking Algorithm

The proposed tracker is formulated within the particle filtering framework [168], and consists of two components: fusion of multiple sparse representations with the proximity constraint and learning the adaptive proximity constraint using online LogDet regularized multiple metric learning algorithm.

4.3.1 Particle Filter

We formulate the proposed tracking algorithm under the framework of sequential Bayesian inference [168]. Let $o_t$ and $l_t$ be the observation and state variable at time $t$, respectively. Once the set of observation $O_t = \{o_t, t = 1, ..., T\}$ up to time $T$ is given, the particle filter approximates the true posterior $P(l_t|O_t)$ using a set of weighted particles $l^i_t, i = 1, ..., n$. Then the tracking result, which is described by the state variable $l_t$, is estimated by,

$$\hat{l}_t = \arg \max_{l^i_t} p(l^i_t|O_t)$$  \hspace{1cm} (4.3.1)

The tracking problem is thus formulated as recursively estimating the posterior probability $p(l_t|O_t)$,

$$p(l_t|O_t) \propto p(o_t|l_t) \int p(l_t|l_{t-1})p(l_{t-1}|O_{t-1})dl_{t-1}$$  \hspace{1cm} (4.3.2)

where $P(o_t|l_t)$ and $P(l_t|l_{t-1})$ denote the observation model and the motion model, respectively. The motion model in [168] is applied in our proposed tracker, and the observation model is defined by the proposed feature fusion model, which will be described in the following subsection.

4.3.2 Fusion of Multiple Sparse Representations with the Proximity Constraint

In the particle filtering based-tracking framework, we have an object template set with multiple visual cues, e.g. color, shape, texture to represent the object in the current frame. Let $y^k$ and $x^k_n$ denote the $k$-th visual cue of the current tracking
result and the $k$-th visual cue of the $n$-th template, respectively. Inspired by the sparse tracker [87], the current tracking result can be sparsely represented by a linear combination of the object templates with an additive noise vector $\varepsilon^k$, i.e.

$$y^k = X^k w^k + \varepsilon^k, k = 1, \cdots, K$$  \hspace{1cm} (4.3.3)

where $w^k \in \mathbb{R}^N$ is the coefficient vector to reconstruct the $k$-th visual cues of the current tracking result using template set $X^k = [x^k_1, \ldots, x^k_N]$, and $N$ is the number of templates.

In (4.3.3), the representations for the visual cues $\{y^k\}_{k=1}^K$ of the current tracking result with respected to the template sets $\{X^k\}_{k=1}^K$ are encoded in the coefficient vectors $\{w^k\}_{k=1}^K$, respectively. Therefore, the feature fusion of different visual cues $\{y^k\}_{k=1}^K$ can be exploited by regularizing the coefficient vectors $\{w^k\}_{k=1}^K$. Here we treat each visual cue representation in (4.3.3) as a sparse pattern of the current tracking result. Then, in order to learn the fused representation among multiple sparse patterns, the optimization model can be defined using the multiple sparse representation framework as follows:

$$\min_{\{w^k\}_{k=1}^K} \sum_{k=1}^{K} \left( \frac{1}{2} \|y^k - X^k w^k\|_2^2 + \lambda_1 \|w^k\|_1 \right) + \lambda_2 \Omega(\{w^k\}_{k=1}^K)$$ \hspace{1cm} (4.3.4)

where the first term is the sparse representations of the current tracking result with multiple cues using $\ell_1$ norm minimization [87], $\lambda_1$ is the trade off between the reconstruction error and the sparseness, $\Omega$ is the regularization function on $\{w^k\}_{k=1}^K$ with parameter $\lambda_2$ for the fusion of multiple sparse representations.

With the goal of modeling the commonalities and diversities among multiple visual cues for object representation, we introduce the regularization function for the fusion of multiple sparse representations in (4.3.4) as follows,

$$\Omega = \sum_{k=1}^{K} \alpha^k \|w^k - w^*\|_2^2$$ \hspace{1cm} (4.3.5)

where $w^*$ is the consensus vector to be learned to reflect the commonalities of representation coefficients $\{w^k\}_{k=1}^K$ of multiple cues, and $\alpha^k$ is the parameter to control the disagreement between $w^k$ and $w^*$. Different from feature concatenation [127]
or joint sparse representation [48] which enforces all representations to be the same or with the same sparsity pattern, the function $\Omega$ in (4.3.5) softly regularizes the representation of each visual cue toward a common consensus, making them similar while allowing them to have some diversities with the consensus, which leads to a more flexible and informative representation for the object. Therefore, both commonalities and diversities among features are exploited for appearance modeling. It also should be noted that with some proper priors on $\{\alpha^k\}_{k=1}^K$, the degree of disagreement of different visual cues with the consensus could be different. A larger (less) $\alpha^k$ will lead to a less (larger) diversity between $w^k$ and $w^*$, which further exploits the diversity of different visual cues.

We provide a better interpretation of (4.3.4) using the regularizer in (4.3.5). Let $w^k = w^* + v^k, k = 1, ..., K$. Then (4.3.4) with the regularizer in (4.3.5) can be rewritten as,

$$
\min_{\{v^k\}_{k=1}^K, w^*} \sum_{k=1}^K \left( \frac{1}{2} \|y^k - X^k(w^* + v^k)\|_2^2 + \lambda_1 \|w^* + v^k\|_1 + \lambda_2 \alpha^k \|v^k\|_2^2 \right)
$$

(4.3.6)

Since $\|w^* + v^k\|_1 \leq \|w^*\|_1 + \|v^k\|_1$ holds, in order to decouple $w^*$ and $v^k$ in $\|w^* + v^k\|_1$ for tractable optimization while explicitly modeling the commonalities and diversities, we relax (4.3.6) by minimizing the upper bound of $\|w^* + v^k\|_1$, i.e. $\|w^*\|_1 + \|v^k\|_1$ as,

$$
\min_{\{v^k\}_{k=1}^K, w^*} \sum_{k=1}^K \left( \frac{1}{2} \|y^k - X^k(w^* + v^k)\|_2^2 + \lambda_1 \|w^*\|_1 + \lambda_1 \|v^k\|_1 + \lambda_2 \alpha^k \|v^k\|_2^2 \right)
$$

(4.3.7)

In (4.3.7), we can clearly see that the representation coefficients $w^*$ of each visual cue are decomposed into $w^*$ and $v^k$, which model the shared commonalities and diversities, respectively. We can also see that the commonality component $w^k$ determines the shared representation pattern of each visual cue. As in [48], we incorporate the proximity constraint into (4.3.7) to favor the joint selection of templates with multiple visual cues which are more representative and similar to the object appearance. Then (4.3.7) can be further developed as,
\[
\min_{\{v^k\}_{k=1}^K, w^*} \sum_{k=1}^K \left( \frac{1}{2} \|y^k - X^k(w^* + v^k)\|_2^2 + \lambda_1 \|d \odot w^*\|_1 + \lambda_1 \|v^k\|_1 + \lambda_2 \alpha^k \|v^k\|_2^2 \right) 
\] (4.3.8)

where \( \odot \) denotes the element-wise multiplication, \( d = \frac{1}{Z}[d_1, \ldots, d_N]^T \in \mathbb{R}^N \), \( d_n = \sqrt{\hat{g}(\{y^k\}_{k=1}^K, \{x^k_n\}_{k=1}^K)} \), \( n = 1, \ldots, N \) encode the proximities between the object and the templates with multiple visual cues, \( \hat{g}(\cdot, \cdot) \) is the learned distance function of two samples with multiple visual cues, and \( Z \) is the normalization factor. Compared to [48] which uses the Euclidean distance to define the proximity constraint, our model adaptively learns the proximity constraint using a novel online multiple metric learning method, which guarantees the closest matching between the object and the representative templates with multiple visual cues so that the commonalities among their representations are more adaptive and more accurate as the object representation. The proposed multiple metric learning algorithm to learn the distance function \( \hat{g}(\cdot, \cdot) \) and the optimization procedure for (4.3.8) will be given in Section 4.3.3 and 4.3.4, respectively.

**Observation Likelihood Function** After the representation coefficients of different visual cues are obtained via solving (4.3.8), the observation likelihood function in (4.3.2) is defined as,

\[
p(a_t|l_t) \propto \exp\left(- \sum_{k=1}^K \|y^k - X^k(w^* + v^k)\|_2^2 \right) 
\] (4.3.9)

The right hand side of this equation denotes the total reconstruction error of the object using the template set with multiple cues.

### 4.3.3 Learning Adaptive Proximity Constraint Using Online LogDet Regularized Multiple Metric Learning

Since the object’s appearance and background change dynamically, a fixed predefined metric may not be able to achieve the closest matching between objects of interest in feature space. Therefore, the metric which defines the proximity constraint should be updated adaptively. To this end, we propose an online LogDet
regularized multiple metric learning to learn the distance function of the proximity constraint in (4.3.8).

Assume that the sample pair with multiple visual cues for metric learning is given with this form \( \{x_1^{k_1}\}_{k_1=1}^{K}, \{x_2^{k_1}\}_{k_1=1}^{K}, y \) where \( \{x_1^{k_1}\}_{k_1=1}^{K} \) and \( \{x_2^{k_1}\}_{k_1=1}^{K} \) are training samples with \( K \) types of visual cues, the subscripts 1 and 2 are the order indexes in the sample pair, and \( y = +1(-1) \) if \( \{x_1^{k_1}\}_{k_1=1}^{K} \) and \( \{x_2^{k_1}\}_{k_1=1}^{K} \) are considered similar (dissimilar). The distance function with \( K \) types of visual cues in (4.3.8) is defined as a linear weighted combination of distance function of each visual cue, i.e.

\[
\hat{g}(\{x_1^{k_1}\}_{k_1=1}^{K}, \{x_2^{k_1}\}_{k_1=1}^{K}) = \sum_{k=1}^{K} \omega^{k} g^{k}(x_1^{k}, x_2^{k})
\]  

(4.3.10)

where \( \omega^{k} \) is the nonnegative weight associated with \( g^{k}(\cdot, \cdot) \), and \( \sum_{k=1}^{K} \omega^{k} = 1 \), and \( g^{k}(x_1^{k}, x_2^{k}) = (x_1^{k} - x_2^{k})^{T} M^{k}(x_1^{k} - x_2^{k}) \) is the distance function for each visual cue with Mahalanobis metric \( M^{k} \) for \( k = 1, ..., K \). The proposed learning strategy aims to learn \( \{\omega^{k}\}_{k=1}^{K} \) and \( \{M^{k}\}_{k=1}^{K} \) such that all the training sample pairs obey the following constraints as well as possible,

\[
\forall(\{x_1^{k_1}\}_{k_1=1}^{K}, \{x_2^{k_1}\}_{k_1=1}^{K}, y) : y = +1 \Rightarrow \sum_{k=1}^{K} \omega^{k} g^{k}(x_1^{k}, x_2^{k}) \leq \theta - \frac{\rho}{2}
\]

(4.3.11)

\[
\forall(\{x_1^{k_1}\}_{k_1=1}^{K}, \{x_2^{k_1}\}_{k_1=1}^{K}, y) : y = -1 \Rightarrow \sum_{k=1}^{K} \omega^{k} g^{k}(x_1^{k}, x_2^{k}) \geq \theta + \frac{\rho}{2}
\]

where \( \theta \) is a predefined threshold. The constraints in (4.3.11) can be further combined as,

\[
\sum_{k=1}^{K} \omega^{k} y(g^{k}(x_1^{k}, x_2^{k}) - \theta + \frac{\rho}{2}) \leq 0
\]

(4.3.12)

The above constraint ensures that all pairs of similar and dissimilar training samples are separated by the margin \( \rho \). Therefore, by putting it into the large-margin learning framework, the loss function of a training sample pair in each visual cue is defined as the square of hinge loss, i.e.

\[
L^{k}(g^{k}(x_1^{k}, x_2^{k}), y) = \frac{1}{2} \max(0, y(g^{k}(x_1^{k}, x_2^{k}) - \theta + \frac{\rho}{2}))^{2}
\]

(4.3.13)

Then the loss function of \( K \) types of visual cues is defined as \( \hat{L} = \sum_{k=1}^{K} \omega^{k} L^{k}(g^{k}(x_1^{k}, x_2^{k}), y) \).

In order to control the model complexity, we adopt the LogDet divergence [23] as
the regularizer, i.e.
\[
R(M^k; M^{k,0}) = tr(M^k(M^{k,0})^{-1}) - \log |M^k(M^{k,0})^{-1}| - m^k
\] (4.3.14)
where \(tr(\cdot)\) is the trace operator, \(\log |\cdot|\) is the log determinant operation, \(m^k\) is the dimension of \(M^k\). Such a regularizer can automatically preserve the positive definiteness of \(M^k\) without the inefficient projection onto positive semi-definite cone, which is suitable for efficient online tracking. Besides, it also can impose some prior information by regularizing the learned metric close to \(M^{k,0}\). Assume that we are given \(T\) training sample pairs, the objective can be defined by the loss function plus the regularizer, i.e.
\[
\min_{\{\omega^k\}_{k=1}^K, \{M^k\}_{k=1}^K} \sum_{k=1}^K R(M^k; M^{k,0}) + \eta \sum_{t=1}^T \sum_{k=1}^K \omega^k L^k(g^k(x_1^{k,t}, x_2^{k,t}), y^t) \quad (4.3.15)
\]
\[
s.t. \sum_{k=1}^K \omega^k = 1, \omega^k \geq 0, k = 1, ..., K
\]
Since all the training sample pairs during the tracking process cannot be obtained at the same time, the model parameters should be learned in an online manner. Inspired by the online multiple kernel classification method reported in [42], we propose an online LogDet regularized multiple metric learning algorithm, in which we employ the Hedging algorithm [26] to learn the weights based on historical information, and then apply the LEGO algorithm [52] to update the metric for each visual cue. In the \(t\)-th trial, the weight and the metric for each visual cue can be estimated as discussed in the following two steps:

**Weight Updating:** By employing the Hedging algorithm, the combination weights are updated as follows,
\[
\tilde{\omega}^{k,t} = \omega^{k,t-1} \beta^{\tau^{k,t}},
\]
\[
\omega^{k,t} = \frac{\tilde{\omega}^{k,t}}{\sum_{k=1}^K \tilde{\omega}^{k,t}}
\] (4.3.16)
where \(\beta \in (0, 1)\) is the discounting parameter, and \(\tau^{k,t}\) equals to 1 if \(y^t(g^{k,t-1}(x_1^{k,t}, x_2^{k,t}) - \theta) \geq 0\), and 0 otherwise. We can see that \(\tau^{k,t} = 1\) indicates that the distance function \(g^{k,t-1}(\cdot, \cdot)\) make a prediction mistake on the sample \(\{x_1^{k,t}, x_2^{k,t}, y^t\}\).
Algorithm 6: Online Multiple Metric Learning

**Input:** Initial Metrics: \( \{M^{k,0}\}_{k=1}^{K} \); Discounting parameter: \( \beta; \eta \)

**Output:** \( \{M^{k,t}\}_{k=1}^{K} \) and \( \{\omega^{k,t}\}_{k=1}^{K}, \ t = 1, \ldots, T \)

**Initialization:** \( \omega^{k}_0 = \frac{1}{k}, \ k = 1, \ldots, K \)

for \( t = 1 \) to \( T \) do

Receive the training sample pair: \( (\{x^{k,t}_1\}_{k=1}^{K}, x^{k,t}_2 \}_{k=1}^{K}, y^t) \)

for \( k = 1 \) to \( K \) do

if \( y^t(g^{k,t-1}(x^{k,t}_1, x^{k,t}_2) - \theta) \geq 0 \) then

\( \tau^{k,t} = 1 \)

Update \( M^k \) using (4.3.18)

else

\( \tau^{k,t} = 0 \)

end

end

Update \( \{\omega^{k,t}\}_{k=1}^{K} \) using (4.3.16)

end
Metric Updating: After the weight updating step, we determine the index set $O$ of the distance function such that $O = \{k', \text{s.t. } y'(g^{k',t-1}(x_{1}^{k',t}, x_{2}^{k',t}) - \theta) \geq 0\}$. By employing the LEGO algorithm [52], the distance metric $M^{k',t}, k' \in O$ is estimated by minimizing the following objective function:

$$\min_{M^{k'}} R(M^{k'}; M^{k',t-1}) + \eta L^{k}(g^{k}(x_{1}^{k,t}, x_{2}^{k,t}), y') \quad (4.3.17)$$

Let $z^{k',t} = x_{1}^{k',t} - x_{2}^{k',t}, s^{t} = (z^{k',t})^T (M^{k',t-1})z^{k',t}$ and $s^{t} = \theta + \frac{y^{t'}}{2}$. The solution to (4.3.17) can be obtained:

$$\bar{s} = \frac{\eta s^{t}s^{t} - 1 + \sqrt{(\eta s^{t}(s^{t})^2 - 1)^2 + 4\eta(s^{t})^2}}{2\eta s^{t}}$$

$$M^{k',t} = M^{k',t-1} - \frac{\eta(s^{t} - s^{t})M^{k',t-1}z^{k',t}(z^{k',t})^T M^{k',t-1}}{1 + \eta(s^{t} - s^{t})(z^{k',t})^T M^{k',t-1}z^{k',t}} \quad (4.3.18)$$

The multiple metric learning algorithm is summarized in Algorithm 6.

Theorem 4.3.1. Assume the norm of each training sample of each visual cue is bounded: $\| x^{k,t} \|_{2}^{2} \leq B$ for $k = 1, ..., K, t = 1, ..., T$, and the optimal metric for all visual cues satisfy $0 \prec M^{k}_{\ast} \prec I, k = 1, ..., K$, then the number of trials that make mistakes by running the proposed metric learning algorithm is bounded as follows:

$$M \leq \frac{8 \cdot \ln(1/\beta)}{\kappa \rho^2 (1 - \beta)} \min_{1 \leq k \leq K} U^{k} + \frac{\ln K}{\kappa (1 - \beta)}$$

where $\kappa = \min\{\omega^{k,t}|t = 1, ..., T, k = 1, ..., K\}$, $U^{k} = \left(1 + \eta \left(\frac{B^2}{4} + \frac{1}{\eta}\right)^2\right) L_{M^{k}_{\ast}} + \left(\frac{1}{\eta} + \left(\frac{B^2}{4} + \frac{1}{\eta}\right)^2\right) R(M^{k}_{\ast}, M^{k}_{0})$ is the loss bound of the LEGO algorithm [52], and $L_{M^{k}_{\ast}} = \sum_{t=1}^{T} \frac{1}{2} \max(0, y'(g_{M^{k}_{\ast}}(x_{1}^{k,t}, x_{2}^{k,t}) - \theta + y^{t'})^2)$

The proof of theorem (4.3.1) is provided in appendix. From theorem (4.3.1), we can see that the number of trials which contributes to prediction mistake is bounded and related to the loss bound of each visual cue, which shows that the multiple metric learning method implicitly incorporates the dependence among multiple metrics.

With the background samples that used for metric learning, the metric learning method also provides more discriminative information for the tracking algorithm, which enhances its robustness to background distracters.
4.3.4 Optimization Procedure

The objective function in (4.3.8) consists of a differential function and a non-differential one and can be efficiently optimized by using the Accelerated Proximal Gradient Method [8]. Let

\[ F(w^*, \{v^k\}_k=1^K) = \sum_{k=1}^K \left( \frac{1}{2} \|y^k - X_k(w^* + v^k)\|^2_2 + \lambda_2 \alpha^k \|v^k\|^2_2 \right) \]

\[ G(w^*, \{v^k\}_k=1^K) = \lambda_1 \sum_{k=1}^K (\|d \odot w^*\|_1 + \|v^k\|_1) \]

where \( F(w^*, \{v^k\}_k=1^K) \) are differential convex function with Lipschitz continuous gradient, while \( G(w^*, \{v^k\}_k=1^K) \) are non-differential convex function. In the \((t + 1)\)-th iteration, with the aggregation \( p^t \) and \( \{q^{k,t}\}_k=1^K \), the solution \( \{v^{k,t+1}\}_k=1^K \) and \( w^{*,t+1} \) are obtained via solving the following minimization problem:

\[
\begin{align*}
\min_{\{v^k\}_k=1^K, w^*} & F(p^t, \{q^{k,t}\}_k=1^K) + \nabla F^T_{p^t}(w^* - p^t) \\
& + \sum_{k=1}^K \nabla F^T_{q^{k,t}}(v^k - q^{k,t}) + \mu_2 \|w^* - p^t\|_2^2 + \\
& \frac{\mu_2}{2} \sum_{k=1}^K \|v^k - q^{k,t}\|_2^2 + \lambda_1 \cdot K \|d \odot w^*\|_1 + \lambda_1 \sum_{k=1}^K \|v^k\|_1
\end{align*}
\]

(4.3.20)

where \( \mu \) is the Lipschitz constant [8]. With some algebraic manipulations, (4.3.20) can be decomposed into the following subproblems:

\[
\begin{align*}
\min_{w^*} & \frac{1}{2} \|w^* - (p^t - \frac{1}{\mu} \nabla F_{p^t})\|_2^2 + \frac{\lambda_1 \cdot K}{\mu} \|d \odot w^*\|_1 \\
\min_{\{v^k\}_k=1^K} & \sum_{k=1}^K \left( \frac{1}{2} \|v^k - (q^{k,t} - \frac{1}{\mu} \nabla F_{q^{k,t}})\|_2^2 + \frac{\lambda_1}{\mu} \|v^k\|_1 \right)
\end{align*}
\]

(4.3.21)

where the partial derivatives of \( F \) with respective to \( p \) and \( \{q^k\}_k=1^K \) at \( p^t \) and \( \{q^{k,t}\}_k=1^K \) are given by \( \nabla F_{p^t} = \sum_{k=1}^K (X_k^T) (X_k(p^t + q^{k,t}) - y^k) \) and \( \nabla F_{q^{k,t}} = (X_k^T) (X_k(p^t + q^{k,t}) - y^k) + 2 \alpha^k \lambda_2 q^{k,t} \). Then the above subproblems can be solved in two steps iteratively:

**Gradient Mapping Step:** Given the aggregation \( p^t \) and \( \{q^{k,t}\}_k=1^K \), the \( w^{*,t+1} \)
and \( \{v^{k,t+1}\}_{k=1}^{K} \) can be updated by (4.3.22) as follows,

\[
p^{t+\frac{1}{2}} = p^{t} - \frac{1}{\mu} \nabla F(p^{t}),
\]

\[
w^{*,t+1}_{n} = \text{sign}(p^{t+\frac{1}{2}}_{n}) \max\left(\frac{d_{n}K\lambda_{1}}{\mu}, 0\right).
\]

\[
q^{k,t+\frac{1}{2}} = q^{k,t} - \frac{1}{\mu} \nabla F(q^{k,t}),
\]

\[
v^{k,t+1}_{n} = \text{sign}(q^{k,t+\frac{1}{2}}_{n}) \max\left(\frac{q^{k,t+\frac{1}{2}}_{n}}{\lambda_{1}} - \frac{\lambda_{1}}{\mu}, 0\right).
\]

for \( n = 1, ..., N, k = 1, ..., K \).

It can be seen that the shrinkage operator for \( w^{*,t+1}_{n} \) in (4.3.22) is closely related to \( d_{n} \), and a smaller \( d_{n} \) which means the corresponding object template is more similar to the object’s appearance will result in a smaller threshold, so that the template will be more likely to be selected for object representation.

**Aggregation Step:** The aggregation \( p^{t+1} \) and \( \{q^{k,t+1}\}_{k=1}^{K} \) can be updated as follows [8]:

\[
p^{t+1} = w^{*,t+1} + \frac{a_{t} - 1}{a_{t+1}}(w^{*,t+1} - w^{*,t})
\]

\[
q^{k,t+1} = v^{k,t+1} + \frac{a_{t} - 1}{a_{t+1}}(v^{k,t+1} - v^{k,t}), k = 1, ..., K
\]

(4.3.23)

where \( a_{t+1} = \frac{1+\sqrt{1+4a_{t}^{2}}}{2} \), and \( a_{0} = 1 \). The overall optimization procedure for (4.3.8) is summarized in Algorithm (7).

**Computational Complexity Analysis** The computational complexity of each iteration in the above optimization algorithm is dominated by the gradient computation. Let \( l \) be the maximum dimension among all visual cues, \( n \) the number of particles, \( K \) the number of visual cues, and \( N \) the number of templates, the computational complexity of the gradient computation is \( O(nKN^{2}l) \). For the online multiple metric learning, each metric update step takes \( O(Kl^{2}) \) computation.

### 4.3.5 Discussion

The proposed tracker fuses multiple sparse representations by exploiting commonalities and diversities among multiple features using sparse pattern decomposition. It is closely related to the \( \ell_{1} \) tracker [87], and provides a more general formulation
Algorithm 7: Optimization Procedure for Problem (4.3.8)

**Input**: Template set \( \{X^k\}_{k=1}^K \), target candidate sample \( \{y^k\}_{k=1}^K \), regularization parameters \( \lambda_1 \) and \( \lambda_2 \), Lipschitz constant \( \mu \), \( \{\alpha^k\}_{k=1}^K \)

**Output**: \( w^* \) and \( \{v^k\}_{k=1}^K \)

**Initialization**: \( p^0 \leftarrow 0, q^{k,0} \leftarrow 0, G^0_p \leftarrow 0, G^0_q \leftarrow 0 \), and \( a_0 = 1 \) repeat

\[
G^t_p = \sum_{k=1}^K (X^k)^T (X^k (p^t + q^{k,t}) - y^k)
\]

\( G^t_q = (X^k)^T (X^k (p^t + q^{k,t}) - y^k) + 2\alpha^k \lambda_2 q^{k,t}, \ k = 1, \ldots, K \)

\[
p^{t+\frac{1}{2}} = p^t - \frac{1}{\mu} G^t_p
\]

\[
q^{k,t+\frac{1}{2}} = q^{k,t} - \frac{1}{\mu} G^t_q, \ k = 1, \ldots, K
\]

\[
w^{n,t+1}_n = \text{sign}(p^{n+\frac{1}{2}}_n) \max \left( \frac{1}{p^{n+\frac{1}{2}}_n} \left| \frac{q^{n+\frac{1}{2}}_n}{\mu} \right| - \frac{d_n K \lambda_1}{\mu}, 0 \right), n = 1, \ldots, N, k = 1, \ldots, K
\]

\[
v^{k,t+1}_n = \text{sign}(q^{k,n+\frac{1}{2}}) \max \left( \frac{1}{q^{k,n+\frac{1}{2}}_n} \left| \frac{w^{n+\frac{1}{2}}_n}{\mu} \right| - \frac{\lambda_1}{\mu}, 0 \right), n = 1, \ldots, N, k = 1, \ldots, K
\]

\[
a_{t+1} = 1 + \sqrt{1 + 4a_t^2}
\]

\[
p^{t+1} = w^{*,t+1} + \frac{a_t - 1}{a_{t+1}} (w^{*,t+1} - w^{*,t})
\]

\[
q^{k,t+1} = v^{k,t+1} + \frac{a_t - 1}{a_{t+1}} (v^{k,t+1} - v^{k,t}), \ k = 1, \ldots, K
\]

until convergence

of the \( \ell_1 \) tracker in term of feature fusion. On the other hand, the proposed tracker differs with other decomposition-based multi-feature sparse trackers [44].

**Relationship with the \( \ell_1 \) tracker [87]**: The proposed tracker is a more general sparse model to fuse multiple features compared with the \( \ell_1 \) tracker. On one hand, if the commonality component is left aside in our tracking model (4.3.8), learning the diversity components of each feature is equivalent to learning multiple \( \ell_1 \) trackers independently, which may lead to diverse representation patterns among multiple features. On the other hand, if all diversity components are ignored in (4.3.8), obtaining the commonality component of all features is the same to learn the \( \ell_1 \) tracker of concatenated features with proximity constraint, which discovers their shared patterns. Therefore, by considering both the diversities and commonalities and fusing multiple \( \ell_1 \) trackers implicitly, the proposed tracking framework is able to exploit different visual cues for appearance modeling so that it is more capable of accounting for different appearance and background changes while taking advantages
of the $\ell_1$ tracker.

**Difference with the multi-task multi-view tracker [44]:** It should be noted that the multi-task multi-view tracker (MTMVT) [44] also uses some decomposition techniques to learn the representation of multiple views of multiple particles. The proposed tracker is totally different from [44] in the following aspects. First, by decomposing the representation matrix into outlier and non-outlier parts, the MTMVT method aims to capture the outlier particles to increase the tracking accuracy and efficiency. But for the multiple views of non-outlier particles, they are strictly restricted to share the same sparsity patterns. That is to say, only the same templates of multiple features from non-outlier particles are allowed to be selected for object representation, which may not be true actually and lead to an inflexible representation scheme. The proposed tracker aims to exploit the commonalities and diversities of all feature patterns in every particle, which is more flexible and more fully unleash the capability of the object templates. Second, the MTMVT method is developed under the robust multi-task learning framework [32] in which the representation matrix is decomposed explicitly using group lasso penalty. The learning model developed in the proposed tracker is different from [32] in which the decomposition formulation is derived from relaxing the regularizer in (4.3.5). Furthermore, as discussed in [49], if non-representative and dissimilar templates with multiple visual cues are jointly selected for sparse representation, background image patches may be decided as the final result. With the same merit of [49], the proposed tracker further incorporates the adaptive proximity constraint into the tracking model to select similar templates for representation. Such issue is not considered in the MTMVT method.
4.4 Implementation Details

4.4.1 Model Update Scheme

In order to adapt to appearance changes and robustness to outliers, we define a reconstruction threshold and similarity threshold for the model updating scheme. As in most sparsity-based trackers [7, 87], each template of multiple visual cues is associated with a weight to indicate the importance of it, and all these weights are updated in each video frame. If the reconstruction error of current tracking result in (4.3.9) is smaller than the reconstruction threshold and the similarity between the current tracking result and the template with the largest weight is smaller than the similarity threshold, then the template with the smallest weight is replaced by the current tracking result. It should be noted that different from the update scheme in the ℓ₁ tracker that updates the template with a single visual cue, once the template in one visual cue of the proposed tracker is updated, the templates of other visual cues are updated simultaneously to maintain the correlation between different visual cues of the same template for commonality learning, and the current tracking results, the template set and some random sampled background image patch are also collected for distance function updating.

4.5 Experiments

In this section, we evaluate the proposed tracker using real videos from publicly available dataset and tracking benchmark.

4.5.1 Experiment Setting

To show the effectiveness of the proposed multi-cue tracking framework, we only use two kinds of hand-crafted features for fusion. For global visual cues, we use HOG [20] to represent the whole object. For local visual cues, the object is divided into 2 by 2 non-overlapped blocks and covariance descriptors [49] are extracted in
### Table 4.1: Center Location Error. The best three results are shown in red, blue and green.

<table>
<thead>
<tr>
<th></th>
<th>Struck</th>
<th>CSK</th>
<th>DFT</th>
<th>OAB</th>
<th>SCM</th>
<th>SemiT</th>
<th>MIL</th>
<th>MTT</th>
<th>LIAPG</th>
<th>ASLA</th>
<th>Ours</th>
<th>COM</th>
<th>EP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basketball</td>
<td>118.4</td>
<td>6.6</td>
<td>18.4</td>
<td>205.5</td>
<td>52.8</td>
<td>328.4</td>
<td>92.2</td>
<td>106.9</td>
<td>137.7</td>
<td>82.6</td>
<td>9.5</td>
<td>123</td>
<td>79.7</td>
</tr>
<tr>
<td>Crowds</td>
<td>4.7</td>
<td>4.4</td>
<td>3.5</td>
<td>8.1</td>
<td>8.3</td>
<td>339</td>
<td>26.3</td>
<td>448.3</td>
<td>315</td>
<td>4.7</td>
<td>3.2</td>
<td>29.6</td>
<td>9</td>
</tr>
<tr>
<td>Football</td>
<td>17.2</td>
<td>16</td>
<td>9.2</td>
<td>72.6</td>
<td>16.2</td>
<td>244.7</td>
<td>12.2</td>
<td>13.4</td>
<td>15</td>
<td>14.9</td>
<td>13.2</td>
<td>13</td>
<td>14.1</td>
</tr>
<tr>
<td>Trellis</td>
<td>7.1</td>
<td>19.2</td>
<td>44.3</td>
<td>98.7</td>
<td>6.9</td>
<td>69.5</td>
<td>72</td>
<td>69</td>
<td>62.3</td>
<td>7.6</td>
<td>7.4</td>
<td>11.7</td>
<td>17.2</td>
</tr>
<tr>
<td>Singer2</td>
<td>174.8</td>
<td>185.7</td>
<td>21.4</td>
<td>188.1</td>
<td>113.9</td>
<td>356.3</td>
<td>22.9</td>
<td>210.1</td>
<td>181.3</td>
<td>175.7</td>
<td>11.7</td>
<td>152</td>
<td>43.9</td>
</tr>
<tr>
<td>Skating1</td>
<td>83.2</td>
<td>8</td>
<td>174.6</td>
<td>43.2</td>
<td>16.5</td>
<td>288.3</td>
<td>139.6</td>
<td>239.8</td>
<td>159.2</td>
<td>60.1</td>
<td>10</td>
<td>102.8</td>
<td>62</td>
</tr>
<tr>
<td>Mountain-bike</td>
<td>9</td>
<td>6.4</td>
<td>154.7</td>
<td>12.2</td>
<td>10.8</td>
<td>240.7</td>
<td>72.6</td>
<td>7.3</td>
<td>8.5</td>
<td>9.2</td>
<td>7.2</td>
<td>122.4</td>
<td>8.3</td>
</tr>
<tr>
<td>Subway</td>
<td>4.1</td>
<td>164.8</td>
<td>3.2</td>
<td>113.2</td>
<td>3.4</td>
<td>105.6</td>
<td>8</td>
<td>166.3</td>
<td>148.1</td>
<td>138.9</td>
<td>6</td>
<td>104.5</td>
<td>6.9</td>
</tr>
<tr>
<td>Face</td>
<td>19.3</td>
<td>12.5</td>
<td>23.8</td>
<td>25.1</td>
<td>13.6</td>
<td>75.1</td>
<td>30.1</td>
<td>21.6</td>
<td>17.8</td>
<td>78</td>
<td>15.1</td>
<td>22</td>
<td>14.7</td>
</tr>
<tr>
<td>Walking</td>
<td>10.7</td>
<td>17.3</td>
<td>28.5</td>
<td>28.7</td>
<td>1.5</td>
<td>11.9</td>
<td>60.2</td>
<td>3.5</td>
<td>4.5</td>
<td>37</td>
<td>4.9</td>
<td>24.7</td>
<td>49.5</td>
</tr>
<tr>
<td>Woman</td>
<td>4.7</td>
<td>207.1</td>
<td>8.2</td>
<td>32.4</td>
<td>8.1</td>
<td>212.2</td>
<td>124.7</td>
<td>137</td>
<td>128.3</td>
<td>139.2</td>
<td>11.8</td>
<td>134.9</td>
<td>113.2</td>
</tr>
<tr>
<td>Bolt2</td>
<td>111.8</td>
<td>11</td>
<td>276.5</td>
<td>7.3</td>
<td>11.1</td>
<td>74</td>
<td>6.6</td>
<td>109.1</td>
<td>61.6</td>
<td>116.9</td>
<td>10.7</td>
<td>44.8</td>
<td>179.7</td>
</tr>
<tr>
<td>Surfer</td>
<td>7.8</td>
<td>175.5</td>
<td>207</td>
<td>110.9</td>
<td>64.7</td>
<td>53.4</td>
<td>5.5</td>
<td>24.6</td>
<td>6.5</td>
<td>17.4</td>
<td>5.5</td>
<td>9.7</td>
<td>4.7</td>
</tr>
<tr>
<td>Bolt</td>
<td>399.3</td>
<td>430</td>
<td>367.7</td>
<td>256.5</td>
<td>456.9</td>
<td>363.7</td>
<td>394.1</td>
<td>499.2</td>
<td>409</td>
<td>375.2</td>
<td>10.8</td>
<td>351.7</td>
<td>122.9</td>
</tr>
<tr>
<td>Shaking</td>
<td>30.1</td>
<td>17.6</td>
<td>26.3</td>
<td>191.6</td>
<td>11.2</td>
<td>275</td>
<td>24</td>
<td>97.3</td>
<td>109.7</td>
<td>22</td>
<td>14.1</td>
<td>65.5</td>
<td>54.2</td>
</tr>
<tr>
<td>David indoor</td>
<td>42.3</td>
<td>17.8</td>
<td>42.5</td>
<td>22.2</td>
<td>4.1</td>
<td>46.8</td>
<td>17.3</td>
<td>32.5</td>
<td>13.8</td>
<td>4.6</td>
<td>5.8</td>
<td>9.7</td>
<td>5.2</td>
</tr>
</tbody>
</table>

### Table 4.2: Overlapping Rate. The best three results are shown in red, blue and green.

<table>
<thead>
<tr>
<th></th>
<th>Struck</th>
<th>CSK</th>
<th>DFT</th>
<th>OAB</th>
<th>SCM</th>
<th>SemiT</th>
<th>MIL</th>
<th>MTT</th>
<th>LIAPG</th>
<th>ASLA</th>
<th>Ours</th>
<th>COM</th>
<th>EP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basketball</td>
<td>0.2</td>
<td>0.71</td>
<td>0.61</td>
<td>0.03</td>
<td>0.47</td>
<td>0.02</td>
<td>0.22</td>
<td>0.2</td>
<td>0.23</td>
<td>0.39</td>
<td>0.47</td>
<td>0.15</td>
<td>0.35</td>
</tr>
<tr>
<td>Crowds</td>
<td>0.71</td>
<td>0.73</td>
<td>0.78</td>
<td>0.56</td>
<td>0.55</td>
<td>0.05</td>
<td>0.62</td>
<td>0.1</td>
<td>0.08</td>
<td>0.66</td>
<td>0.75</td>
<td>0.26</td>
<td>0.58</td>
</tr>
<tr>
<td>Football</td>
<td>0.55</td>
<td>0.56</td>
<td>0.66</td>
<td>0.34</td>
<td>0.49</td>
<td>0.15</td>
<td>0.59</td>
<td>0.58</td>
<td>0.56</td>
<td>0.54</td>
<td>0.62</td>
<td>0.57</td>
<td>0.6</td>
</tr>
<tr>
<td>Trellis</td>
<td>0.62</td>
<td>0.49</td>
<td>0.36</td>
<td>0.14</td>
<td>0.68</td>
<td>0.2</td>
<td>0.25</td>
<td>0.22</td>
<td>0.2</td>
<td>0.8</td>
<td>0.65</td>
<td>0.6</td>
<td>0.64</td>
</tr>
<tr>
<td>Singer2</td>
<td>0.04</td>
<td>0.04</td>
<td>0.63</td>
<td>0.05</td>
<td>0.17</td>
<td>0.03</td>
<td>0.52</td>
<td>0.04</td>
<td>0.04</td>
<td>0.75</td>
<td>0.06</td>
<td>0.38</td>
<td>0.38</td>
</tr>
<tr>
<td>Skating1</td>
<td>0.31</td>
<td>0.5</td>
<td>0.14</td>
<td>0.4</td>
<td>0.47</td>
<td>0.08</td>
<td>0.13</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.58</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>Mountain-bike</td>
<td>0.71</td>
<td>0.71</td>
<td>0.3</td>
<td>0.64</td>
<td>0.68</td>
<td>0.23</td>
<td>0.46</td>
<td>0.75</td>
<td>0.74</td>
<td>0.73</td>
<td>0.76</td>
<td>0.32</td>
<td>0.71</td>
</tr>
<tr>
<td>Subway</td>
<td>0.66</td>
<td>0.2</td>
<td>0.74</td>
<td>0.17</td>
<td>0.73</td>
<td>0.29</td>
<td>0.66</td>
<td>0.07</td>
<td>0.16</td>
<td>0.19</td>
<td>0.56</td>
<td>0.27</td>
<td>0.54</td>
</tr>
<tr>
<td>Face</td>
<td>0.73</td>
<td>0.8</td>
<td>0.69</td>
<td>0.66</td>
<td>0.8</td>
<td>0.57</td>
<td>0.6</td>
<td>0.75</td>
<td>0.32</td>
<td>0.78</td>
<td>0.69</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>Walking</td>
<td>0.53</td>
<td>0.47</td>
<td>0.41</td>
<td>0.37</td>
<td>0.82</td>
<td>0.47</td>
<td>0.29</td>
<td>0.79</td>
<td>0.76</td>
<td>0.38</td>
<td>0.56</td>
<td>0.34</td>
<td>0.33</td>
</tr>
<tr>
<td>Woman</td>
<td>0.73</td>
<td>0.2</td>
<td>0.76</td>
<td>0.49</td>
<td>0.67</td>
<td>0.11</td>
<td>0.16</td>
<td>0.17</td>
<td>0.16</td>
<td>0.15</td>
<td>0.54</td>
<td>0.07</td>
<td>0.15</td>
</tr>
<tr>
<td>Bolt2</td>
<td>0.4</td>
<td>0.51</td>
<td>0.01</td>
<td>0.68</td>
<td>0.58</td>
<td>0.2</td>
<td>0.74</td>
<td>0.03</td>
<td>0.17</td>
<td>0.01</td>
<td>0.53</td>
<td>0.22</td>
<td>0.24</td>
</tr>
<tr>
<td>Surfer</td>
<td>0.47</td>
<td>0.03</td>
<td>0.03</td>
<td>0.04</td>
<td>0.17</td>
<td>0.48</td>
<td>0.27</td>
<td>0.56</td>
<td>0.38</td>
<td>0.67</td>
<td>0.43</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>Bolt</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.04</td>
<td>0.02</td>
<td>0.06</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.56</td>
<td>0.04</td>
<td>0.22</td>
</tr>
<tr>
<td>Shaking</td>
<td>0.35</td>
<td>0.58</td>
<td>0.64</td>
<td>0.01</td>
<td>0.69</td>
<td>0.01</td>
<td>0.43</td>
<td>0.04</td>
<td>0.08</td>
<td>0.47</td>
<td>0.63</td>
<td>0.11</td>
<td>0.16</td>
</tr>
<tr>
<td>David indoor</td>
<td>0.24</td>
<td>0.41</td>
<td>0.3</td>
<td>0.39</td>
<td>0.73</td>
<td>0.25</td>
<td>0.43</td>
<td>0.3</td>
<td>0.54</td>
<td>0.75</td>
<td>0.64</td>
<td>0.54</td>
<td>0.74</td>
</tr>
</tbody>
</table>
Figure 4.2: Qualitative results on some typical frames including some challenging factors. (a) Illumination and cluttered background. (b) Deformation and occlusion.

each block. The parameters are set as follows: $\lambda_1$ and $\lambda_2$ in (4.3.8) is set as 0.001 and 0.033, respectively, and $\alpha_k, k = 1, ..., 5$ are all set as 3. $\eta$ in (4.3.8) is set as 0.125. We set the $\theta$ and $\rho$ in (4.3.12) to be 1.5 and 1. And the discounting parameter $\beta$ in (4.3.16) is set as 0.9. The number of templates maintained in the template set is set as 12. The initial Mahalanobis metric of each visual cues are learned by the ITML algorithm [23].

4.5.2 Evaluation on Publicly Available Sequences

In this section, we evaluate the proposed tracking algorithm by conducting experiments on sixteen publicly available image sequences\(^1\), which covers different kinds of challenging factors including cluttered background, pose, non-rigid deformation, illumination. We compare the proposed tracking algorithm with other ten state-of-the-art trackers which include the multi-cue trackers: OAB [33], SemiT [34], MIL [4], sparse representation based-trackers: MTT [158], L1APG [7], ASLA [54],

SCM [162] and other state-of-the-arts methods: Struck [37], DFT [95], CSK [40]. The source codes for other compared trackers are provided by the authors of these papers. To validate the effectiveness of modeling the diversity among multiple features, we implemented the tracker which only considers the commonality and fuses multiple features using concatenation, and we call this method **COM** for short.

To better understand the importance of the adaptive proximity constraint in our proposed tracking algorithm, we implemented the tracker which is also with proximity constraint but it is defined by Euclidean distance for comparison. We call this method **EP** for short. For fair comparison, all trackers are set to be with the same initialization parameters.

**Quantitative Comparison**

Following [112, 162, 134, 123, 111, 110, 159], two evaluation metrics are adopted for quantitative comparison: center location error and VOC overlapping rate. Center location error (CLE) is the Euclidean distance between the centers of bounding box and ground-truth, and the VOC overlapping rate is define as $\frac{\text{area}(B_T \cap B_G)}{\text{area}(B_T \cup B_G)}$ where $B_G$ and $B_T$ are the bounding boxes of the tracker and ground-truth. Table 4.1 and Table 4.2 illustrate the video-by-video comparison results of the evaluated trackers in terms of center location error and overlapping rate, respectively. The proposed tracking algorithm ranks in the top three in most videos in terms of center location error and overlapping rate, and it achieves much better performance than sparsity-based trackers with single feature, e.g. MTT, L1APG, ASLA, and SCM. This is because the proposed tracker is able to fuse multiple sparse representations with multiple features and dynamically exploit complementarity among multiple features to handle different variations. In addition, different from other sparse trackers, e.g. L1APG which only utilize target information and ignore interclass proximity between foreground and background, the proposed online multiple metric learning algorithm effectively incorporates proximity information into the tracking frame-

---

2The last two columns are for self-comparison and do not participate in performance ranking.
work, which further enhances the tracking performance. Compared with existing trackers with multiple features, e.g. OAB, SemiT and MIL, the proposed tracker also achieves much better performance. This is because the fusion of multiple sparse representations can simultaneously utilize the commonality and diversity among features and take advantages of the robustness of sparse representation while boosting-based methods such as OAB only consider one aspect, i.e. diversity of weak classifiers to fuse classifier scores for visual tracking. Both the EP method and the proposed tracker achieve a better overall performance than the the COM method, which suggests that exploiting the diversity among different feature representations can increase the tracking accuracy. The comparison between the proposed tracker and the EP method suggests that learning an adaptive metric for template matching is more effective in dealing with changing appearance and background, compared with predefined and fixed Euclidean proximity metric.

**Qualitative Comparison**

From the qualitative results on some typical frames as shown in Fig. 4.2, we can see that the proposed tracker is more stable under large appearance variation and cluttered background. This is attributed to two aspects. First, the diversity of the sparse pattern enables different templates from different features to be activated for sparse representation, which exploits the complementarity of multiple features; the commonality among features ensures different features of the sample template to be fused together for representation, which utilizes their relationship for appearance modeling. Therefore, the proposed trackers explicitly model these two components, leading to a more informative appearance model. In spite of being with the same global and local features as the proposed tracker’s, the COM method, which enforces all representation patterns to be the same, can not achieve satisfactory performance under some challenging factors, such as deformation (Bolt#30). This is because enforcing all patterns to be the same makes it much less flexible for each local feature to model the target deformation. By incorporating the diverse sparse pattern,
the proposed tracking model enables each feature from local parts to have different representation to handle irregular non-rigid deformation adaptively. Second, the multiple metric learning method adaptively incorporates discriminability from multiple features for appearance modeling. As shown in the typical frames of Trellis, Shaking and Skating1 of Fig. 4.2(a), we can see that the proposed tracker is more stable than the EP method under cluttered background. This is because the proposed metric learning method, which aims to maximize the interclass distance and to minimize the intraclass distance using multiple features, make it more easy for the targets to be separated from background by the tracker.

4.5.3 Evaluation on Visual Tracking Benchmark

As in [157, 110], we also compared the proposed trackers with the other ten trackers using One-Pass-Evaluation (OPE) on the tracking benchmark [128] which contains totally 51 videos covering numerous challenging factors such as cluttered background, low resolution, illumination and so on. It should be noted that the Struck and the SCM methods are the top 2 trackers based on OPE as reported in [128]. Two kinds of widely accepted metrics, precision plot and success plot are used for benchmark evaluation [128]. For the success plot which shows the percentage of success frames over a given threshold varying from 0 to 1, the performance score is computed based on the area under the AUC curve. For the precision plot which shows the ratio of frames whose CLE is within a given threshold changing from 0 to 50, the performance score is the precision value on the threshold 20 as in [128].

Precision Plots and Success Plots

Fig. 4.1 demonstrate the success plots and the precision plots of the One-Pass-Evaluation (OPE) on the whole set and the subsets of sequences contain different challenging factors, repetitively. We can see that the proposed tracker achieves the best overall performance using the two metrics. Our tracker also ranks in the top two among the compared trackers using both precision plots and success plots for
8 out of 11 challenging subsets. Moreover, the proposed tracker demonstrates excellent performance on the subsets of sequences which covers cluttered background, low resolution, in-plane/out-plane rotation. This is because the effectiveness of the fusion scheme and adaptive proximity constraint learned by the proposed metric learning method make it less sensitive to cluttered background and enable the target to be distinguished from low resolution images with multiple visual cues. As observed in [49], the fusion of local covariance descriptors also enables it to describe the object appearance under pose variation.

**Survival Curve**

To provide a cumulative rendition of the quality of the tracker on a set of videos and avoid the risk of being trapped in the peculiarity of the single video instance in tracker performance evaluation, similar to the ALOV++ evaluation [100], we evaluate the trackers with the survival curves using the F-scores computed from the tracking results of each tracker on each video. The F-score for each video can be computed as $2 \cdot (\text{precision} \cdot \text{recall})/(\text{precision} + \text{recall})$, where precision $= n_{tp}/(n_{tp} + n_{fp})$, recall $= n_{tp}/(n_{tp} + n_{fn})$, and $n_{tp}, n_{fp}, n_{fn}$ denote the number of true positives, false positives, false negatives in a video. For each tracker, after the F-score of each video in the video set is obtained, we sort the sets of videos in the descending order and plot the survival curve of F-scores with respect to the sorted videos as shown in Fig. 4.4. We can see that the proposed tracker achieves the best overall performance compared to the other ten trackers with highest average F-score on all videos, which further validates the robustness of the proposed tracking algorithm.

**Comparison with Other Proposed Methods**

We further compare the performance of the proposed methods in this chapter with that of the proposed methods in previous chapters. The comparison is based on the popular tracking video dataset [128]. The One Pass Evaluation based on the success
Figure 4.3: The precision plots and the success plots of One-Pass-Evaluation on the subsets of sequence contains different challenging factors. (a) Fast motion and background clutter. (b) Motion blur and deformation. (c) In-plane and out-plane rotation. (d) Low resolution and occlusion. (e) Illumination and out of view. (f) scale variation. The ranking result and performance score of each tracker is shown in the legend of each figure. The precision plots and the success plots of One-Pass-Evaluation of all video sequence are shown in the last two figures of (f).
plot and the precision plot are adopted as evaluation metrics. Since the proposed three methods are based on sparse representation, the $\ell_1$ tracker [7] are used as the baseline methods. Figure 4.5 shows the comparison results of the proposed methods and the baseline method. The CDMT method performs better than the proposed joint sparse tracker and the feature learning-based tracker, which shows that by exploiting the commonality and diversity in representations of multiple features, and imposing the proximity constraint with background information, the tracking performance can be improved. Although the RJDFT use the multiple sparse representations with different features for target representations, different from the CDMT and the RJSRFFT method, it only considers the independency of different representations and does not consider the correlation of different sparse representations in feature fusion. Therefore, the RJDFT method does not outperform the CDMT and RJSRFFT methods, which illustrates that exploiting the relationship of different features for target representation is very important. We also can see that all the proposed methods outperform the L1 tracker, which demonstrates the importance and effectiveness of feature learning and fusion with multiple visual cues.
4.6 Conclusion

In this chapter, we have proposed a multiple sparse representation framework with adaptive proximity constraint for multi-cue visual tracking. By explicitly modeling both the commonality and diversity among features, the proposed tracker is able to construct a more flexible and informative appearance model than other multi-cue sparse trackers. Experimental results on publicly available data show that the proposed tracker performs favorably against other ten state-of-the-arts tracking methods which include multi-cue trackers, sparse trackers and the top two trackers as reported in [128], under appearance variations due to pose, illumination and occlusion.
Chapter 5

Conclusion

5.1 Summary

Visual tracking is an important and active research topic in computer vision community because of its wide range of applications, e.g., intelligent video surveillance, human computer interaction and robotics. Although it has been extensively studied in the last two decades, it still remains to be a challenging problem due to many appearance variations caused by occlusion, pose, illumination and so on. Because different visual cues (features) are insensitive to different variations and the use of multiple features could compensate the limitations of each feature and provide better performance, to tackle these challenges from appearance variations, a promising direction is to properly use multiple visual cues(features) in appearance modeling. Along this direction, we propose appearance models for multi-cue visual tracking which addresses several research issues in feature learning and fusion.

Since the extracted raw features from tracking samples may be contaminated and suffer the loss of discriminability, in order to extract uncontaminated discriminative features from multiple visual cues, this thesis proposes a robust joint discriminative feature learning framework for multi-cue appearance modeling. The proposed feature learning framework is capable of removing corrupted/contaminated features, imposing discriminability into the learned features and exploiting the consistent and feature-specific discriminative information from multiple visual cues, which is
a joint optimal framework to take advantage of both generative and discriminative approaches to describe the tracked object more accurately and distinguish the foreground and background more discriminatively. Experiment results on fifteen challenging videos show its effectiveness for feature extraction in multi-cue visual tracking.

With the extracted visual features, a key issue in feature fusion is how to dynamically select and fuse appropriate features for more robust appearance modeling. To address this issue, considering that fusion on feature level contains more information than that in score level, this thesis proposes a novel feature-level fusion model based on joint sparse representation for multi-cue visual tracking in which feature selection, fusion and representation are performed jointly so that unreliable features which do not share the same sparsity pattern as reliable features are removed and reliable features are fused for representation. Extensive experiment results on synthetic data and real videos demonstrate the effectiveness of both the unreliable feature detection scheme and the feature fusion model.

Different features extracted from the same object should share some commonalities in their representations while each feature should also have some diversities to reflect its complementarity, and the benefit of utilizing the commonality and diversities of multiple features are also well demonstrated in many classification/recognition tasks. Different from existing sparse representation-based multi-cue visual tracking algorithms which only explicitly model the commonality in multiple representations of different visual cues, this thesis proposes a novel multiple-sparse-representation model which explicitly models the commonality and diversity in the representations of multiple visual cues for feature fusion, which lead to a more informative and accurate representations with multiple features. To facilitate the close matching between the tracked target and the template, a novel online metric learning algorithm is proposed to adaptively incorporate the proximity constraint, which further enhance the tracking accuracy. Extensive experiment results on sixteen challenging videos and tracking benchmark demonstrate the effectiveness of both the feature
fusion model and the multiple metric learning algorithm, and the superiority of the proposed tracker.

5.2 Future work

Although the algorithms proposed in this thesis have achieved convincing results in some challenging videos and the visual tracking benchmark, there is a long way to go for developing a practical visual tracking systems and some issues may be further investigated.

- **Real-Time Tracking** Since the proposed algorithms in this thesis are sparsity-based methods with high dimensional feature descriptors, they cannot achieve real-time speed. In order to meet the requirement from real-world application, several strategies could be further exploited to increase the tracking speed. First, more efficient optimization algorithm can be developed to solve the sparsity-regularized optimization problem. Second, effective dimension reduction techniques can be exploited to reduce the feature dimension. Third, more efficient mathematic manipulations, such as generating negative samples by circulant shifts can be incorporated into implementation.

- **Integrated with detection module for long-term tracking** Most existing tracking algorithms focus on short-term tracking and may not well handle the tracking failure cases. For practical long term tracking, due to the large appearance change, the tracking error may be accumulated to be large. In addition, the tracked target may be out of view in the camera. Both two cases will lead to tracking failure. Therefore, recovery from such tracking failure and recapturing the target based on the previous tracking results are crucial for practical visual tracking system. Integrating tracking with a detection module will help to alleviate the drift problems and recapture the tracked target once the tracker encounters a failure case. Detection during tracking is different from conventional detection algorithm, and existing detection algorithm can-
not be directly integrated into visual tracking. First, the detector should be learned online in order to be adapted to appearance variation during tracking. Second, not all the training samples adopted for learning the detector are reliable. Outlier sample decision scheme should be developed to eliminate the outlier samples. Third, detection is used to detect a specified object but not a category of object. That is to say, the detector can distinguish the tracked object from other objects in the same category. The background information is required for learning the detector in order to alleviate the background distraction problem. Developing a framework to integrate detection and tracking is an important problem for further study.

- **Tracking across non-overlapping camera** With growing installation of video surveillance camera, more and more non-overlapping cameras have been adopted for resource saving and broader field of view. Due to the appearance variation and the inconsistency between different cameras, tracking human across non-overlapping cameras is more challenging than single-camera tracking. In recent years, significant progress has been made in person re-identification. The goal of person re-identification is to match persons across multiple cameras. Most of existing re-identification algorithms are based on image-to-image matching, which cannot well utilize the temporal information in each camera, and incorporating tracking into person re-identification framework provides temporal information in each camera to person re-identification for set-to-set matching, while person re-identification algorithm helps to re-locate the target in different cameras. Exploiting the relationship between the tracking and person re-identification can provide complementary information to each other. It may be a promising direction for further exploration.
Appendices

Proof of Proposition 3.4.1

Proof. The Hessian matrix $\mathcal{H} \in \mathbb{R}^{(2 \cdot N \cdot K) \times (2 \cdot N \cdot K)}$ for function $F$ in (4.3.11) is a block diagonal matrix with $B_k$ as the $k$-th block, where

$$B_k = \begin{bmatrix}
\frac{\partial^2 F}{\partial r^k \partial r^k} & \frac{\partial^2 F}{\partial r^k \partial s^k} \\
\frac{\partial^2 F}{\partial s^k \partial r^k} & \frac{\partial^2 F}{\partial s^k \partial s^k}
\end{bmatrix}
\in \mathbb{R}^{(2 \cdot N) \times (2 \cdot N)}$$

(A1)

Assume that $\Lambda^k = \text{diag}(\lambda_1^k, \ldots, \lambda_N^k)$ is a diagonal matrix whose elements are the eigenvalues for $B_k$, and $Z^k$ is a matrix with eigenvectors of $B_k$ as column vectors. That is to say, $B_k Z^k = Z^k \Lambda^k$.

Let $Z = \text{diag}(Z^1, \ldots, Z^K)$, and $\Lambda = \text{diag}(\Lambda^1, \ldots, \Lambda^K)$, then $\mathcal{H} Z = \text{diag}(B_1 Z^1, \ldots, B_K Z^K) = \text{diag}(Z^1 \Lambda^1, \ldots, Z^K \Lambda^K) = \text{diag}(Z^1, \ldots, Z^K) \text{diag}(\Lambda^1, \ldots, \Lambda^K) = Z \Lambda$. That is to say, the eigenvalues of $\mathcal{H}$ are the same as its diagonal block $B_1, \ldots, B_K$. Hence,

$$\lambda_{\text{max}}(\mathcal{H}) = \max\{\lambda_{\text{max}}(B_k) | k = 1, \ldots, K\}$$

(A2)

$$= \max\{2 \cdot \lambda_{\text{max}}(\mathcal{K}(X^k, X^k)) | k = 1, \ldots, K\}$$

where $\lambda_{\text{max}}(\mathcal{H})$ and $\lambda_{\text{max}}(B_k)$ are maximum eigenvalue of $\mathcal{H}$ and $B_k$, respectively.

The proof is done.
Proof of Theory 4.3.1

Proof. With the assumption in Theorem 4.3.1, let $L_{Mk}$ denote the loss incurred by series of the $k$-th metric after $T$ trials, i.e., $L_{Mk} = \sum_{t=1}^{T} L^k(g^{k,t}(x_{1,t}^{k}, x_{2,t}^{k}), y^t)$, then according to theorem 2.5 in [52], we have $L_{Mk} \leq U^k$. And if the distance function of the $k$-th visual cue $g^{k,t}(\cdot, \cdot)$ makes a mistake at trial $t$, then $L^k(g^{k,t}(x_{1,t}^{k}, x_{2,t}^{k}), y^t) \geq \frac{\rho^2}{8}$.

So we have,

$$\sum_{t=1}^{T} \tau^{k,t} \leq \frac{8}{\rho^2} L_{Mk} \leq \frac{8}{\rho^2} U^k$$

Following the analysis in [26], we have

$$\sum_{t=1}^{T} \sum_{k=1}^{K} \omega^{k,t-1} \tau^{k,t} \leq \frac{\ln(1/\beta)}{(1-\beta)} \min_{1 \leq k \leq K} \sum_{t=1}^{T} \tau^{k,t} + \frac{\ln K}{(1-\beta)}$$

Since the distance functions in the trial $t$ make a mistake if and only if $\sum_{k=1}^{K} \omega^{k,t-1} \tau^{k,t} > 0$, and $\sum_{t=1}^{T} I(\sum_{k=1}^{K} \omega^{k,t-1} \tau^{k,t} > 0) \leq (1/\kappa) \sum_{t=1}^{T} \sum_{k=1}^{K} \omega^{k,t-1} \tau^{k,t}$, we have

$$M = \sum_{t=1}^{T} I(\sum_{k=1}^{K} \omega^{k,t-1} \tau^{k,t} > 0)$$

$$\leq (1/\kappa) \sum_{t=1}^{T} \sum_{k=1}^{K} \omega^{k,t-1} \tau^{k,t}$$

$$\leq \frac{\ln(1/\beta)}{\kappa (1-\beta)} \min_{1 \leq k \leq K} \sum_{t=1}^{T} \tau^{k,t} + \frac{\ln K}{\kappa (1-\beta)}$$

$$\leq \frac{8 \cdot \ln(1/\beta)}{\kappa \rho^2 (1-\beta)} \min_{1 \leq k \leq K} U^k + \frac{\ln K}{\kappa (1-\beta)}$$

$\square$
Bibliography


attributes for real-time visual tracking. In *Proc. CVPR*, pages 1090–1097,
2014.


via label information for multi task object tracking. In *Proc. ICME*, pages 1–6,
2014.

1997.

[27] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking


CURRICULM VITAE

Academic qualifications of the thesis author, Mr. LAN Xiangyuan:

- Received the degree of Bachelor of Engineering in Computer Science and Engineering from South China University of Technology, China, June 2012.

August 2016