USC–SIPI REPORT #440

A DEEP LEARNING APPROACH TO ONLINE SINGLE AND MULTIPLE OBJECT TRACKING

By

Weihao Gan

May 2018

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
USC Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

A DEEP LEARNING APPROACH TO ONLINE SINGLE AND MULTIPLE OBJECT TRACKING

by

Weihao Gan

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

May 2018

Contents

List of Tables

List of Figures

Abstract

Online object tracking is one of the fundamental computer vision problems. It is commonly used in real world applications such as traffic control and safety in video surveillance, autonomous vehicle, robotic navigation, medical imaging, etc. It is a very challenging problem due to multiple time-varying attributes in video sequences. One widely adopted online object tracking framework is tracking-by-detection (TBD), where tracking is treated as a detection problem. This strategy exploits the spatial information of the image content. In this research, we investigate two different kinds of tracking problems: single object tracking (SOT) and multiple object tracking (MOT). First, we attempt to achieve online single object tracking using both spatial and motion cues with two novel methods. Second, from the proposed SOT technique, we build an online multiple object tracking system with advanced model update and matching.

First, we develop a new method, called the ”temporal prediction and spatial refinement (TPSR)” tracker, to integrate spatial and temporal cues effectively. The TPSR tracking system consists of three cascaded modules: pre-processing (PP), temporal prediction (TP) and spatial refinement (SR). Illumination variation and shaking camera movement are two challenging factors in a tracking problem. They are compensated in the PP module. Then, a joint region-based template matching (TM) and pixel-wised optical flow (OF) scheme is adopted in the TP module, where the switch between TM and OF is conducted automatically. These two modes work in a complementary manner to handle different foreground and background situations. Finally, to overcome the drifting error arising from the TP module, the bounding box location and size are finetuned using the local spatial information of the new frame in the SR module.

Next, we apply the deep neural network architecture to the online object tracking problem. The proposed method is called ”Motion-Guided Convolutional Neural Network (MGNet) Tracker”, which is built upon the multi-domain convolutional neural network (MDNet) with two innovations: 1) adoption of a motion-guided candidate selection (MCS) scheme based on a dynamic prediction model, and 2) usage of a RGB-plus-motion 5-channel input to the convolutional neural network (CNN). For the former, a dynamic motion model is adopted to estimate the probability distribution of candidate's location, width and height. As a result, the MGNet can generate candidates more accurately and efficiently. For the latter, we add the horizontal and vertical optical flow fields to the original RGB three channels to form a 5-channel input so that the motion information is exploited explicitly rather than implicitly by the CNN. We compare the performance of the MGNet, the MDNet and several state-of-the-art online object trackers on the OTB and the VOT benchmark datasets, and demonstrate that the temporal motion correlation between any two consecutive frames in videos can be more effectively captured by the MGNet via extensive performance evaluation.

Finally, we start to explore the multiple object tracking (MOT) system based on the CNN single object tracker. The proposed method is called ”Online CNN-based Multiple Object Tracking with Enhanced Model Updates and Identity Association”. This method treats the MOT problem as an online tracking problem, rather than the global optimization framework. There are three major components in this tracking system: 1) a system platform built upon multiple CNN single object trackers in MOT environment; 2) the proposed advanced online update strategy including incremental and refresh update mode; 3) a confirmation process for identity matching based on multiple level feature representations. We evaluate our proposed framework on the commonly used multiple object tracking dataset - MOTchallenge, and rank the top 1 position in accuracy/precision/IDswitch/Fragment among all the online MOT tracking methods. Extensive experiments show that the proposed online update strategy is crucial to train an accurate target tracker and control the error drifting in the future.

Chapter 1

Introduction

1.1 Significance of the Research

One of the main goals of computer vision is to enable computers to replicate the basic functions of human vision, such as object recognition, motion perception and scene understanding. To achieve the goal of intelligent motion perception, much effort has been spent on visual object tracking, which is one of the most important and challenging research topics in computer vision. The tracking problem can be divided into two different subproblems: single object tracking (SOT) and multiple objects tracking (MOT).

Let us first look at the problem definition of the single object tracking problem. In order to clearly understand it, we would like to first give a brief definition of this problem: given the initial location status $\mathbf{x}_0$ of one single target, the goal is to estimate the target location status $\mathbf{x}_t$ at frame t without accessing to the information in the future frames ($t' > t$). One visual object tracking example is shown in Figure 1.1. The target location is initialized in the first frame with green bounding box and the tracker needs to track it in the future frame. Red bounding box indicates the tracking result.

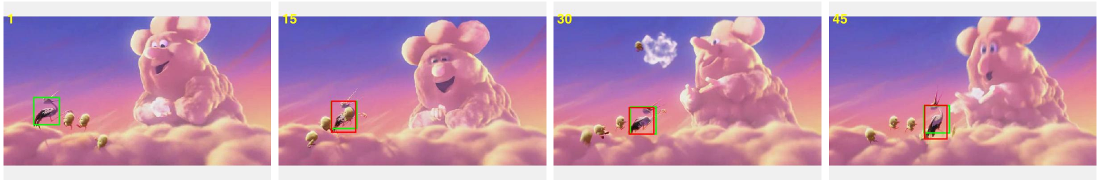


Figure 1.1: Online visual object tracking task example. Green: ground truth labelling. Red: Tracker prediction results.

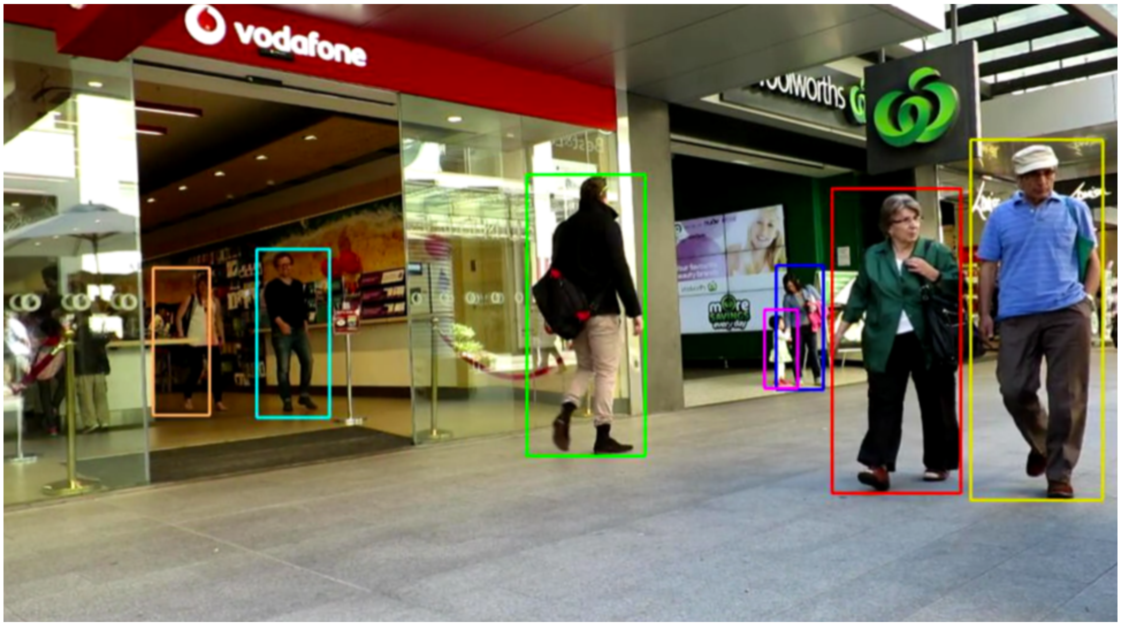


Figure 1.2: Illustration of the idea of multiple object tracking.

As for multiple object tracking (MOT), there are several different aspects on the definition comparing the SOT:

- Target of interest. In SOT, the target is general object while the focus of MOT is pedestrian (Figure 1.2).

- Identity assignment. During the tracking, one important thing is to make sure the identity of each target consistent.

- Initialization from detection. In SOT, system uses the ground truth location of the target to initialize the tracker. While in MOT, the available information is the still image pedestrian detection results.

- No ”online” constraint. Because of the availability of the detection results from all frames, it is possible to globally optimize the detections to form the complete target trajectory.

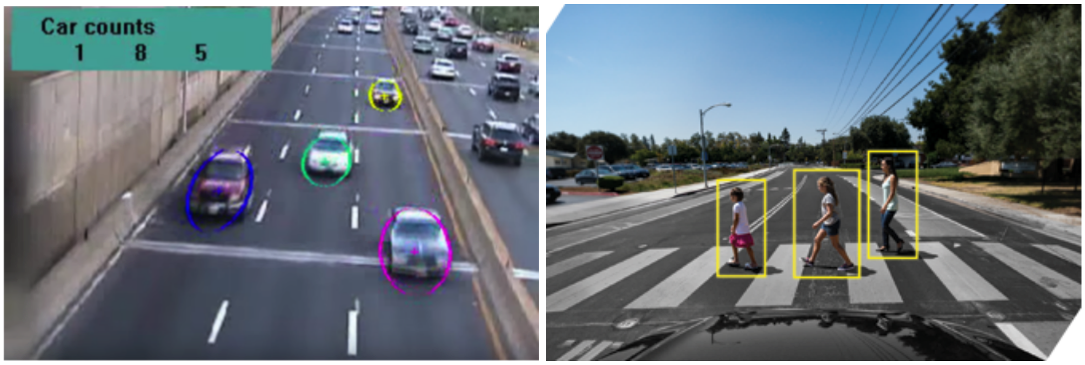


Figure 1.3: Real world applications with visual object tracking.

Many research topics, such as object detection, image/video segmentation and behavior understanding, are all related to visual object tracking. Also, this topic finds many real world applications such as intelligent traffic control in video surveillance, robotics and autonomous vehicle navigation, human computer interaction, medical imaging, etc. For example in the left image of Figure 1.3, it shows that by tracking and counting the number of vehicles, people can control the traffic of different streets. The right image shows the application in autonomous vehicle that by tracking and detecting the movement of pedestrian, the vehicle can make a decision of driving or stopping.

As we can see in Figure 1.3, the tracking forms of the car and pedestrian are different, which are ellipse and bounding box, respectively. There are also some other forms to represent the location of a target, as shown in Figure 1.4. The most commonly used form is still the bounding box representation. Contour form requires a pixel-wise accuracy of the target, which is highly related to object segmentation. Articulated block focuses on part tracking, while interest point focuses on single pixel tracking. In this dissertation, we evaluate all our work and other methods in bounding box form.

Despite extensive research on this topic, visual object tracking is still a very difficult problem. There are three major challenges: (1) Complex object appearance change, (2) Partial and fully occlusion and (3) Generic object representation.

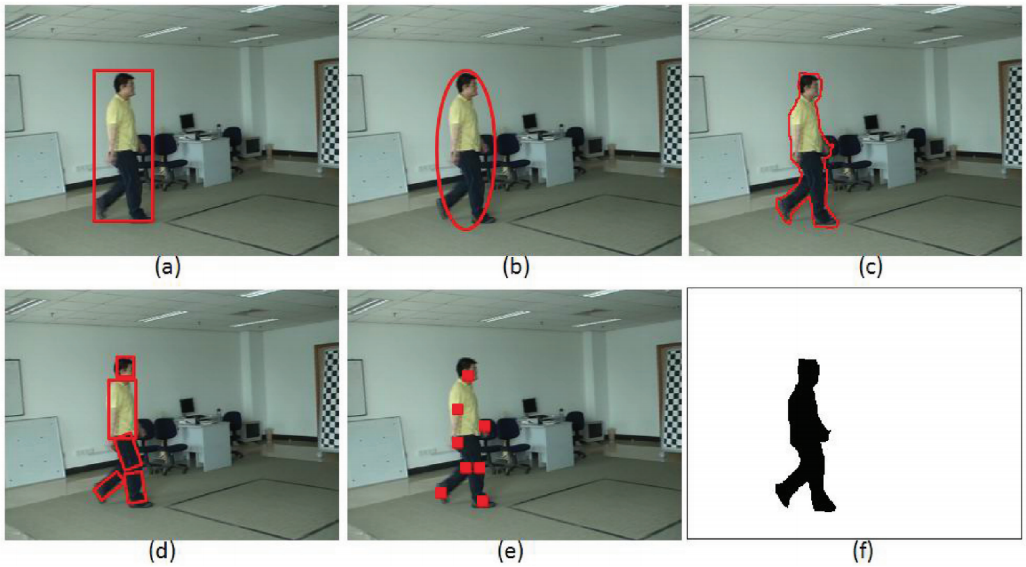


Figure 1.4: Illustration of object tracking forms. (a) bounding box, (b) ellipse, (c) contour, (d) articulated block, (e) interest point, (f) silhouette.

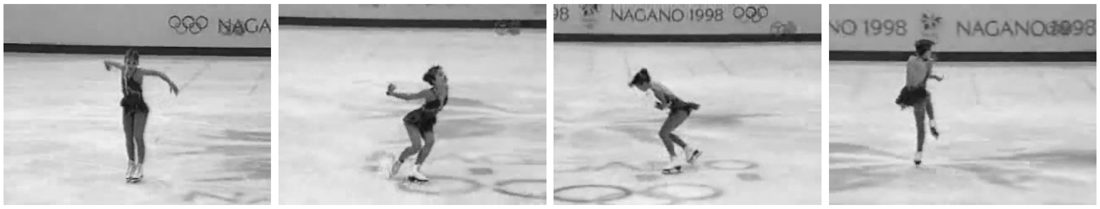


Figure 1.5: Visual object tracking challenging: appearance change.

- The first one is in handling complex object appearance changes caused by many factors of the target, such as scale variation, shape deformation, illumination change, complicated motion, background clusters, etc. For example in Figure 1.5, we can see the pose of the skater changes along the time. In this case, it is extremely difficult for the tracker to find a fixed representation of the target during the tracking. Therefore, tracking system needs to dynamically update their model along the time.

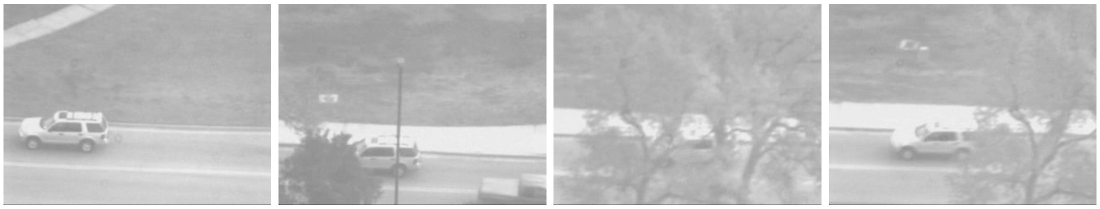


Figure 1.6: Visual object tracking challenging: partial and fully occlusion.

- The second challenge is partial and fully occlusion. One example is shown in Figure 1.6. In this scenario, tracker cannot find a complete target region (SUV) since it is occluded by other region. Usually, the detection/tracking score of occlusion is very low and the system should be able to differentiate the situations between wrong location and fully occlusion. Also, the tracker must have the ability to re-detect the target in the future. If the problem becomes the MOT, the occlusion and interaction between targets will be more severe.

- Besides of the complex object appearance change and occlusion, another challenge in this tracking problem is how to find the generic target representation when a new video is given. The goal of this problem is to track any given object in any video, no matter it is a car, pedestrian, toy, or other objects. We cannot build a target-specific tracker. For example, a pedestrian detector/tracker definitely fails when the target is a car. Therefore, the only opportunity for the tracker to determine the generic representation of the target is the initialization in the first frame. At this step, the tracking system needs to learn the feature of the target and build a model to distinguish this target from the background region. In Figure 1.7, it shows the diversity of different target contents in a commonly used visual object tracking benchmark dataset.

As we can see, visual object tracking plays a significant role in computer vision applications and the diversity of the visual content makes it challenging and difficult.

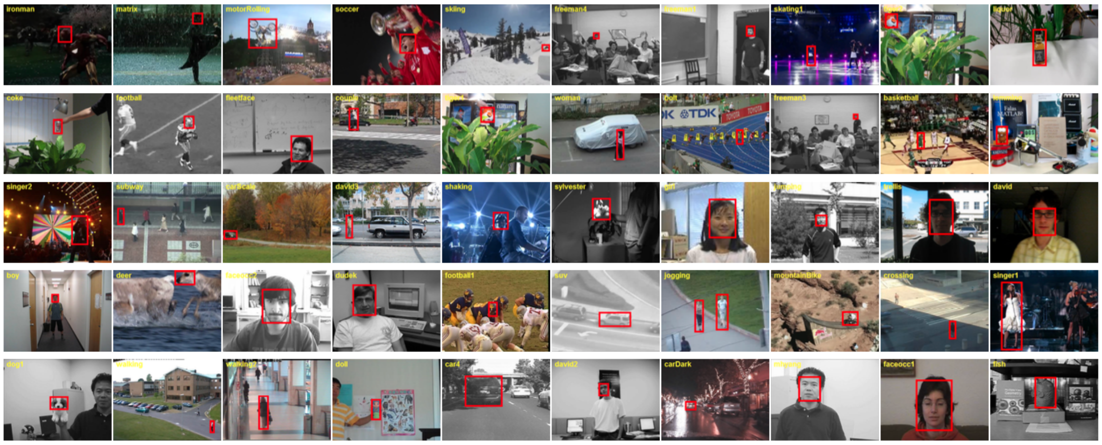


Figure 1.7: Visual object tracking challenging: different objects.

Our goal is to propose a robust object tracker, which can handle with different attributes. For the single object tracking problem, we propose two methods (TPSR and MGNet), trying to solve this problem from a new angle which combines both of the spatial and temporal motion information together. For the multiple object tracking, we extend our CNN-based single object tracker into MOT environment with advanced model update and matching.

1.2 Methodology in SOT

A lot of research progress has been made in online object tracking in recent years. Extensive experiments have been conducted in evaluating various proposed tracking methods against a benchmarking dataset in the literature [55, 93, 112, 113]. Generally speaking, existing trackers can be divided into two categories: traditional visual trackers and convolutional neural network (CNN) trackers.

1.2.1 Traditional Visual Trackers

Before the deep neural network architecture, most traditional tracking algorithms fall into either generative or discriminative approaches. These different methods are centered on two basic components: the object representation scheme and the matching mechanism.

For object representation, a holistic template matching scheme was considered in [2, 14, 75], where the absolute pixel-wise differences in two regions, one in the previous frame (also called the reference frame) and the other in the current frame, were summed up. The minimum that gives the smallest sum of absolute differences (SAD) determines the new object location. The sparse representation was adopted recently to address the large computational burden for matching in the image domain and object's appearance change e.g. [8, 46, 47, 76, 77, 108, 114, 128]. The part-based representation was proposed in [1, 48, 115] to deal with occlusion. Adam *et al.* [1] represented an object using a grid of fragments, and the new object position is obtained by fragments' voting. Jia *et al.* [48] proposed to divide an object into smaller patches by a regular grid. Besides template matching, people also paid attention to local feature descriptors, such as histograms of oriented gradients (HOG) [21], Haar-like features [107] and local scale-invariant features (SIFT) [70].

The matching mechanism depends on the object representation scheme. One widely adopted mechanism is tracking-by-detection, where tracking is treated as a detection task. Variants of online boosting classifiers were chosen in several tracking-by-detection papers [4, 5, 36], where the input features can be patch histograms or local feature descriptors. Recently, tracking-by-detection methods use robust loss functions [66, 74], semi-supervised learning [37, 89, 125], multiple-instance learning [5, 122], or kernel-based support vector machine classification [10, 30, 38, 105]. Also, a cascaded probabilistic tracking approach [123] was presented to solve the tracking problem by using supervised dictionary learning.

One main challenge of the tracking-by-detection scheme is that object's appearance change along time. Kwon *et al.* [61] proposed a framework to combine multiple observation and motion models to handle this problem. Fradi *et al.* [32] exploited both the temporal and spatial cues in the human tracking. Some methods [38, 39, 44, 62] focused on object's short-term behavior while others put emphasis on its long term behavior [49, 64, 72, 86, 100]. Another method proposed in [43] solved the tracking problem by exporting long term and short term modules to achieve the state-of-the-art performance. Research has also been done on trajectory reasoning [45, 65]. Multiple component trackers using color, texture and illumination, respectively, were developed in [65]. The most confident tracker that maximizes the robustness score was selected by analyzing forward and backward trajectories.

1.2.2 Convolutional Neural Network (CNN) Trackers

CNNs have demonstrated their outstanding representation power in a wide range of computer vision applications [35, 56, 101]. AlexNet [56] brought significant performance improvement in image classification by training a deep CNN with a large-scale dataset. R-CNN [35] applies a CNN to an object detection task.

For visual tracking, only a limited number of tracking algorithms using the representations from CNNs have been proposed so far [28, 42, 69, 109]. [69] proposes an online learning method based on a pool of CNNs. However, it suffers from lack of training data to train deep networks and its accuracy is not particularly good compared to the methods based on hand-craft features. A few recent approaches [42, 109] transfer pretrained CNN features into a classification model, but the representation may not be very effective due to the fundamental difference between classification and tracking tasks.

The multi-domain network (MDNet) tracker [85] shows a significant performance gain with deep neural network architecture. It pretrains the network using a set of videos with tracking ground truth annotations to obtain a generic representation for an arbitrary new sequence. The network is composed of two parts - shared layers and domain specific layers, where domains correspond to individual tracking sequences and each domain has a separate branch for binary classification. After training, a generic representation in the shared layers across all domains is obtained. The tracking is performed by sampling target candidates around the previous target state, evaluating them on the CNN, and identifying the sample with the maximum score.

1.3 Methodology in MOT

Tracking-by-detection strategy is the most commonly used idea in various tracking tasks. It shows the impressive performance improvement thanks to the development of the objet detectors. Therefore, in MOT problem, the most popular benchmark dataset - MOTchallenge [63, 78], follows this fashion by directly providing all the targets detection results in all the frames. In this scenario, all the target initialization locations are not labelled by human ground truth, but purely depend on the detection information. Then the task is to link all the detections of one individual object together and form one trajectory (ID assignment). In this dataset, it focuses on the multiple pedestrian tracking. In order to build the complete trajectories of different targets, existing MOT solutions can be roughly categorized into global optimization and online methods.

1.3.1 Global Optimization Methods

Methods in this category [13, 59, 82, 118] focus on minimizing the total energy cost from all the target trajectories. These methods utilize all the detections of whole frames together to link fragmented trajectories due to occlusion. In order to build an more accurate energy affinity measure, idea of ”tracklet” - several consecutive frames, is exploited to extract the spatial and temporal features of the target. Short tracklets are generated by linking the detections and the tracklets are globally associated to build the complete trajectory of the target. Therefore, many global optimization techniques have been proposed, such as graph cut [102,103] and flow network [87,126]. However, the performance of the global optimization methods is still limited under some situations, such as long-term occlusion and missing detection. There is no correct detected bounding box for both of these cases, which increase the difficulty in distinguishing different objects along the time. Moreover, most of the global optimization methods access to the detections for the entire sequence beforehand, and also processing all the datas requires huge computation due to the iterative associations for generating globally optimized tracks. Thus, it is impossible to apply them to real-time applications.

1.3.2 Online Methods

Online methods [6,12,91,96] can be applied to real-time applications because they build each trajectory in a frame-by-frame fashion. In this case, the location and identity of one target are determined in current frame without accessing to the information in the future frames. However, since it is difficult to handle inaccurate (or even absent) detections of occluded objects, online methods tend to produce fragmented trajectories and to drift under occlusion. Therefore, the most challenging task in this kind of methods is to find an accurate feature representation to link the detections to the previous tracks.

1.4 Contributions of the Research

In this dissertation, we investigate the tracking system from SOT and MOT. For SOT problem, two methods are proposed: temporal prediction and spatial refinement (TPSR) tracking system and motion-guided CNN tracker. Both of them exploit the idea of combining spatial and temporal motion cues on tracking problem, but with different frameworks and contributions. For MOT, we extend the CNN single object tracker into the MOT environment. The proposed model update strategy and online ID matching process are the keys to guarantee the tracking performance.

1.4.1 Temporal Prediction and Spatial Refinement (TPSR) Tracking System

- We study many the state-of-the-art tracking-by-detection trackers and find out the drawbacks of this scheme. We observe the advantages of using temporal information in this tracking problem. TPSR is the first work that integrates both of these two cues to build a robust visual object tracker.

- In the temporal prediction (TP) module, the use of template matching (TM) and optical flow predictor (OF) jointly for robust temporal prediction of the bounding box between the reference and the current frame in the TP module is new. By examining the trajectory of the target, we automatically switch the system between these two modes.

- In the spatial refinement (SR) module, the refinement of bounding box's location and size using the spatial domain information of the new frame in the SR stage is also novel. It offers an effective mean to control cumulative errors due to temporal prediction alone.

1.4.2 Motion Guided Convolutional Neural Network (MGNet) Tracker

- The use of dynamic motion model to generate the correct candidate regions is essential for tracking since if the candidates are incorrect, it is impossible to locate the target successfully. On another hand, an accurate target location estimation also reduces the number of candidates and speeds up the tracking process.

- It is the first time that the spatial RGB and temporal optical flow are combined together as the network inputs to show the discriminative power of the tracking system. Optical flow map indicates the motion vector for each pixel between two adjacent frames, which provides important movement and segmentation cues for target localization.

1.4.3 Online Multiple Object Tracking Using CNN Tracker with Advanced Model Update and Matching

- An online MOT framework based on single CNN tracker is introduced. Each target is associated with one unique multi-domain network tracker [85]. In this MOT pipeline, we introduce the target-in and target-out strategy to add and remove target trackers efficiently.

- The proposed online update scheme is important to build an accurate single object tracker in MOT environment. The incremental update strategy is used for the visible target with consecutively successful tracking. The aggressive update strategy is targeted to handle the recaption scenario after the occlusion.

- A multi-level online learned feature representation scheme is proposed to confirm the correct ID to the targets. One is the observation cue from the feature response from different layers in the network. Another one is the motion cue. These two cues are combined together and weighted by a collision factor to assign the correct ID to each target.

1.5 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, we first present several challenging object tracking datasets in SOT and MOT to reveal the significance and difficulties of the problems. Also, some important tools and key techniques in this topic will be described. The proposed Temporal Prediction and Spatial Refinement (TPSR) tracking system will be introduced in Chapter 3. In Chapter 4, we propose an motion guided CNN tracker, which integrates both of spatial and temporal motion cues into one unique multi-domain learning network. In Chapter 5, the proposed online MOT framework will be explained consisting of the flowchart, model update and online matching. In Chapters 3-5, we will explain the detailed methodologies, experimental results and conclusions. Finally, we will summarize this work and discuss future research directions in Chapter 6.

Chapter 2

Visual Object Tracking

In this chapter, we would like to give more inside background of visual object tracking problem. First, several visual object tracking datasets are provided and we will give a brief description and discussion of them. Then, the evaluation methodology will be described. Finally, we will explain several important tools and techniques in this field.

2.1 Datasets

2.1.1 Tracking Datasets in Surveillance Cameras

At early ages of visual object tracking, people collected the sequences from surveillance cameras for performance evaluation. There are two main datasets.

- VIVID dataset [20]

  The domain of interest of this VIVID tracking dataset is tracking ground vehicles from airborne sensor platforms. It is a subset of public release data collected under the DARPA VIVID program. In selecting evaluation video clips, the goal has been to offer a representative sample of object resolution, contrast, pose and degree of occlusion, in both visible and thermal IR imagery. The original video clips are avi movie files encoded via motion-jpeg. All videos have been decoded into sequences of jpeg image files to remove any potential variability due to use of different decoders.

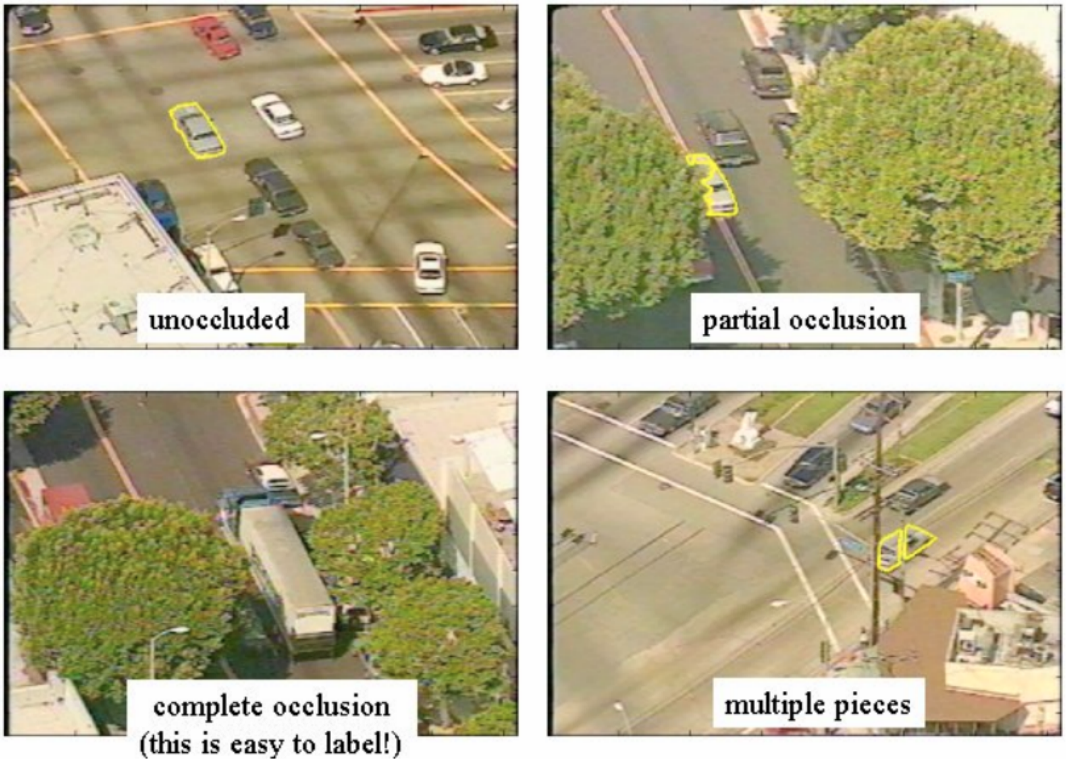


Figure 2.1: Examples of VIVID object tracking dataset.

In this dataset, there are only 8 video sequences for evaluation. One example is shown in Figure 2.1. We can see it focuses on vehicle tracking, which includes several different attributes, such as partial/fully occlusion and multiple pieces.

- CAVIAR dataset [31]

  This dataset is originally targeted for pedestrian detection. However, since it contains video sequences annotated with both target position and activities, it is also used for pedestrian tracking in early year. Altogether, there are 28 video clips containing about 26500 labeled frames. The resolution is half-resolution PAL standard (384 x 288 pixels, 25 frames per second) and compressed using MPEG2. One video example is illustrated in Figure 2.2.

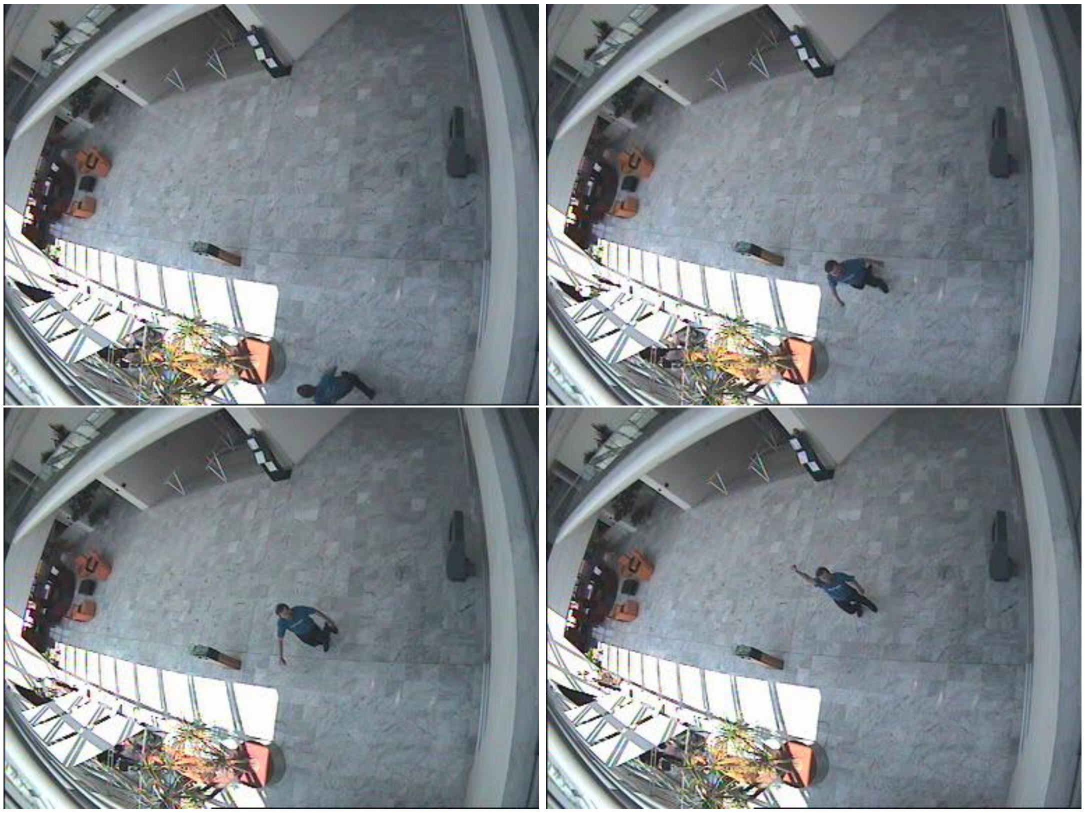


Figure 2.2: Examples of CAVIAR object tracking dataset.

Here, we can see these two early datasets are targeting two specific objects, vehicle and pedestrian, respectively, which are not very suitable for visual object tracking evaluation.

2.1.2 Benchmark Datasets with Generic Objects

- Object tracking benchmark dataset (OTB) [112, 113]

  Comparing to those surveillance sequences, such as VIVID and CAVIAR, whose target objects are usually humans or small cars and the background is usually static, the object tracking benchmark datasets (OTB50, OTB100) [112, 113] are targeted on generic objects. In 2013, OTB50 was proposed, which contains 50 fully annotated sequences with 11 attributes. OTB100 expands the sequences

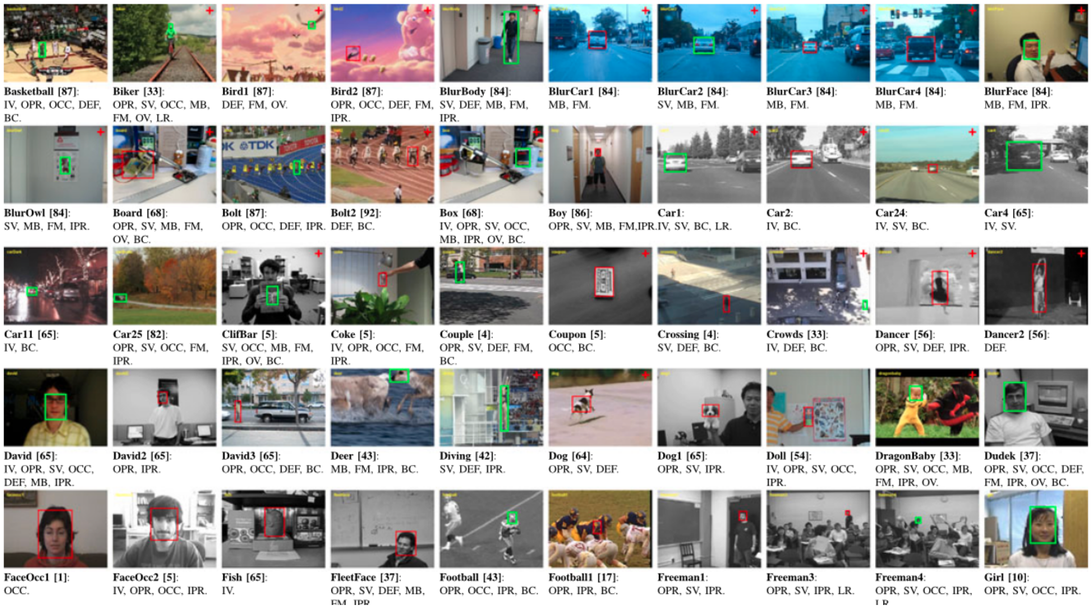

Basketball [87]: IV, OPR, OCC, DEF, BC.
Biker [33]: OPR, SV, OCC, MB, FM, OV, LR.
Bird1 [87]: DEF, FM, OV.
Bird2 [87]: OPR, OCC, DEF, FM, IPR.
BlurBody [84]: SV, DEF, MB, FM, IPR.
BlurCar1 [84]: MB, FM.
BlurCar2 [84]: SV, MB, FM.
BlurCar3 [84]: MB, FM.
BlurCar4 [84]: MB, FM.
BlurFace [84]: MB, FM, IPR.
BlurOwl [84]: SV, MB, FM, IPR.
Board [68]: OPR, SV, MB, FM, OV, BC.
Bolt [87]: OPR, OCC, DEF, IPR.
Bolt2 [92]: DEF, BC.
Box [68]: IV, OPR, SV, OCC, MB, IPR, OV, BC.
Boy [86]: OPR, SV, MB, FM,IPR.
Car1: IV, SV, BC, LR.
Car2: IV, BC.
Car24: IV, SV, BC.
Car4 [65]: IV, SV.
Car11 [65]: IV, BC.
Car25 [82]: OPR, SV, OCC, FM, IPR.
ClifBar [5]: SV, OCC, MB, FM, IPR, OV, BC.
Coke [5]: IV, OPR, OCC, FM, IPR.
Couple [4]: OPR, SV, DEF, FM, BC.
Coupon [5]: OCC, BC.
Crossing [4]: SV, DEF, BC.
Crowds [33]: IV, DEF, BC.
Dancer [56]: OPR, SV, DEF, IPR.
Dancer2 [56]: DEF.
David [65]: IV, OPR, SV, OCC, DEF, MB, IPR.
David2 [65]: OPR, IPR.
David3 [65]: OPR, OCC, DEF, BC.
Deer [43]: MB, FM, IPR, BC.
Diving [42]: SV, DEF, IPR.
Dog [64]: OPR, SV, DEF.
Dog1 [65]: OPR, SV, IPR.
Doll [54]: IV, OPR, SV, OCC, IPR.
DragonBaby [33]: OPR, SV, OCC, MB, FM, IPR, OV.
Dudek [37]: OPR, SV, OCC, DEF, FM, IPR, OV, BC.
FaceOcc1 [1]: OCC.
FaceOcc2 [5]: IV, OPR, OCC, IPR.
Fish [65]: IV.
FleetFace [37]: OPR, SV, DEF, MB, FM, IPR
Football [43]: OPR, OCC, IPR, BC.
Football1 [17]: OPR, IPR, BC.
Freeman1: OPR, SV, IPR.
Freeman3: OPR, SV, IPR, LR.
Freeman4: OPR, SV, OCC, IPR, LR
Girl [10]: OPR, SV, OCC, IPR.

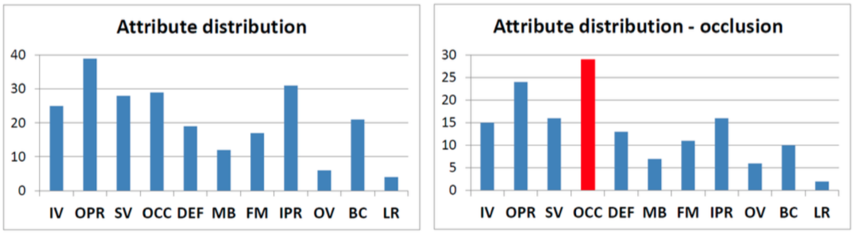
Figure 2.3: Examples of OTB benchmark dataset.

collected in OTB50 to include 100 target objects in the benchmark dataset. To facilitate a fair performance evaluation, OTB collects and annotates diversity of the commonly used tracking targets, including human, face, vehicle, toy, animals, etc, as shown in Figure 2.3.

Another feature provided with OTB dataset is the attribute distribution. Evaluating trackers is difficult because many factors can affect the tracking performance. For better evaluation and analysis of the strength and weakness of tracking approaches, OTB categorizes the sequences by annotating them with the 11 attributes shown in Table 4.8. The attributes distribution in the dataset is shown in Figure 2.4(a). It also shows that one sequence is often annotated with several attributes, which can be even more challenging. For example in Figure 2.4(b), the OCC subset contains 29 sequences which can be used to analyze the performance of trackers to handle occlusion.

Attr	Description
IV	Illumination Variation - the illumination in the target region is significantly changed.
SV	Scale Variation - the ratio of the bounding boxes of the first frame and the current frame is out of the range $[1/t_s, t_s], t_s > 1$ (t_s=2).
OCC	Occlusion - the target is partially or fully occluded.
DEF	Deformation - non-rigid object deformation.
MB	Motion Blur - the target region is blurred due to the motion of target or camera.
FM	Fast Motion - the motion of the ground truth is larger than t_m pixels (t_m=20).
IPR	In-Plane Rotation - the target rotates in the image plane.
OPR	Out-of-Plane Rotation - the target rotates out of the image plane.
OV	Out-of-View - some portion of the target leaves the view.
BC	Background Clutters - the background near the target has the similar color or texture as the target.
LR	Low Resolution - the number of pixels inside the ground-truth bounding box is less than t_r (t_r=400).

Table 2.1: List of attributes annotated to test sequences. The threshold values used in the dataset are also shown.



(a) (b)

Figure 2.4: (a) Attribute distribution of the entire testset, and (b) the distribution of the sequences with occlusion (OCC) attribute.

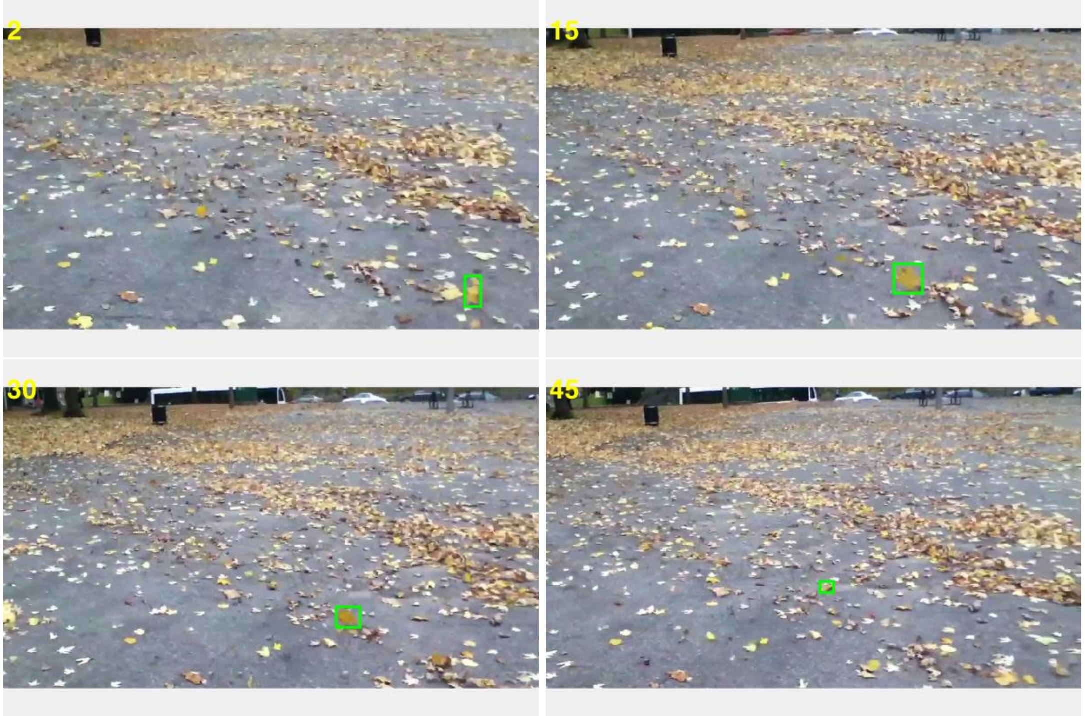


Figure 2.5: Leave example of visual object tracking challenging dataset.

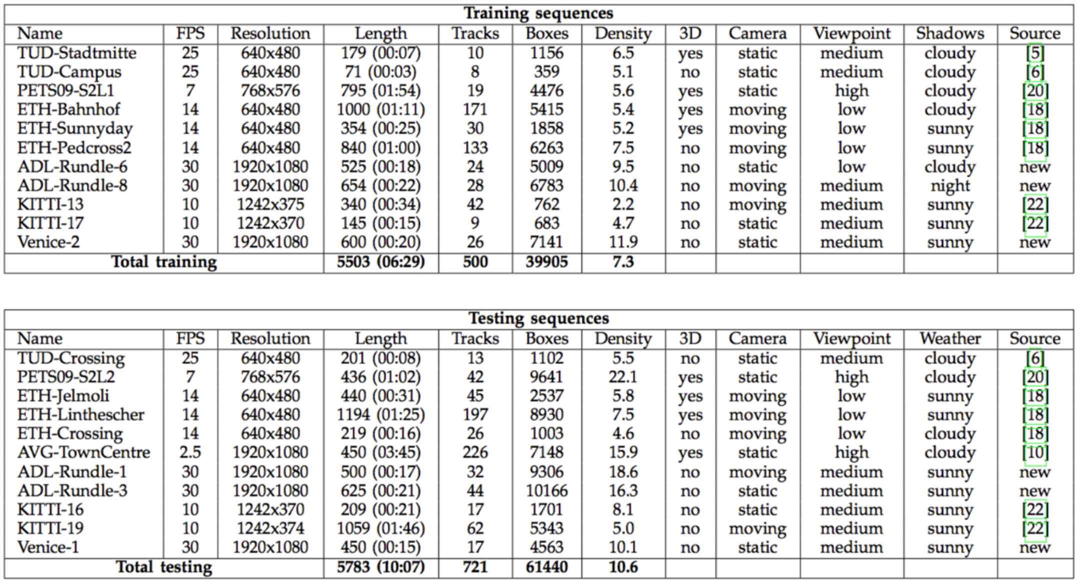
- Visual object tracking challenging dataset (VOT) [54]

  Besides of OTB dataset, another popular dataset is visual object tracking (VOT) challenging [54]. This challenging is held every year since 2013 and all the testing sequences are public released after the competition. The VOT2015 dataset contains 60 sequences showing various objects in challenging backgrounds. The sequences were annotated by the VOT committee using rotated bounding boxes in order to provide highly accurate ground truth values.

  Honestly speaking, this dataset has some extremely difficult sequences that almost all the existing tracking solutions cannot work well on them. One example is shown in Figure 2.5, where the target is a piece of leave. We can see it is very challenging since of the noise from complex background clusters.

2.1.3 Multiple Object Tracking Benchmark Dataset (MOT)

The multiple object tracking benchmark [63] is formed as a challenge competition from 2015. It focuses on multiple pedestrian tracking in a crowed street environment. The most difficult part in this topic is the occlusion/interaction between targets. In 2015, the first version of the dataset was proposed. In total, there are 11 training and 11 testing video sequences. Here is the statistic of the dataset in Figure 2.6.



Training sequences

Name	FPS	Resolution	Length	Tracks	Boxes	Density	3D	Camera	Viewpoint	Shadows	Source
TUD-Stadtmitte	25	640x480	179 (00:07)	10	1156	6.5	yes	static	medium	cloudy	[5]
TUD-Campus	25	640x480	71 (00:03)	8	359	5.1	no	static	medium	cloudy	[6]
PETS09-S2L1	7	768x576	795 (01:54)	19	4476	5.6	yes	static	high	cloudy	[20]
ETH-Bahnhof	14	640x480	1000 (01:11)	171	5415	5.4	yes	moving	low	cloudy	[18]
ETH-Sunnyday	14	640x480	354 (00:25)	30	1858	5.2	yes	moving	low	sunny	[18]
ETH-Pedcross2	14	640x480	840 (01:00)	133	6263	7.5	no	moving	low	sunny	[18]
ADL-Rundle-6	30	1920x1080	525 (00:18)	24	5009	9.5	no	static	low	cloudy	new
ADL-Rundle-8	30	1920x1080	654 (00:22)	28	6783	10.4	no	moving	medium	night	new
KITTI-13	10	1242x375	340 (00:34)	42	762	2.2	no	moving	medium	sunny	[22]
KITTI-17	10	1242x370	145 (00:15)	9	683	4.7	no	static	medium	sunny	[22]
Venice-2	30	1920x1080	600 (00:20)	26	7141	11.9	no	static	medium	sunny	new
Total training			**5503 (06:29)**	**500**	**39905**	**7.3**					

Testing sequences

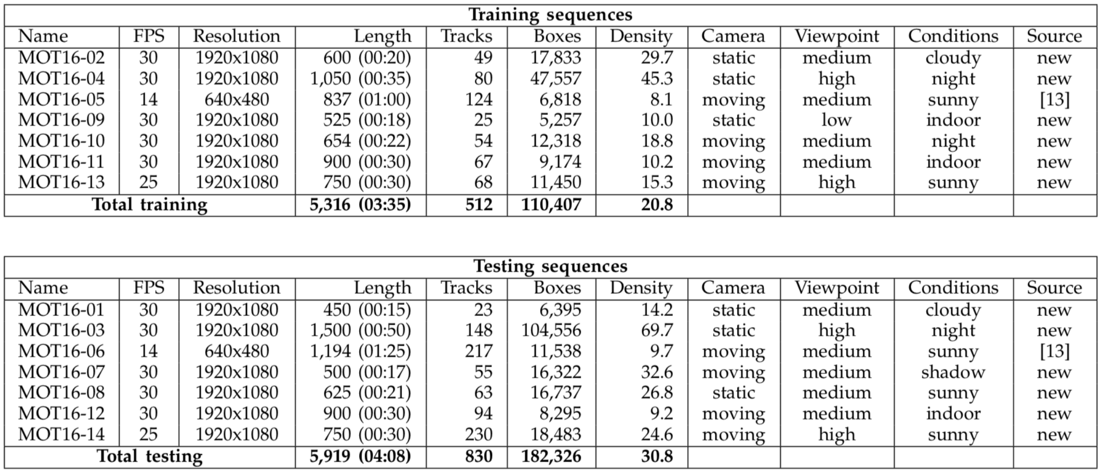
Name	FPS	Resolution	Length	Tracks	Boxes	Density	3D	Camera	Viewpoint	Weather	Source
TUD-Crossing	25	640x480	201 (00:08)	13	1102	5.5	no	static	medium	cloudy	[6]
PETS09-S2L2	7	768x576	436 (01:02)	42	9641	22.1	yes	static	high	cloudy	[20]
ETH-Jelmoli	14	640x480	440 (00:31)	45	2537	5.8	yes	moving	low	sunny	[18]
ETH-Linthescher	14	640x480	1194 (01:25)	197	8930	7.5	yes	moving	low	sunny	[18]
ETH-Crossing	14	640x480	219 (00:16)	26	1003	4.6	no	moving	low	cloudy	[18]
AVG-TownCentre	2.5	1920x1080	450 (03:45)	226	7148	15.9	yes	static	high	cloudy	[10]
ADL-Rundle-1	30	1920x1080	500 (00:17)	32	9306	18.6	no	moving	medium	sunny	new
ADL-Rundle-3	30	1920x1080	625 (00:21)	44	10166	16.3	no	static	medium	sunny	new
KITTI-16	10	1242x370	209 (00:21)	17	1701	8.1	no	static	medium	sunny	[22]
KITTI-19	10	1242x374	1059 (01:46)	62	5343	5.0	no	moving	medium	sunny	[22]
Venice-1	30	1920x1080	450 (00:15)	17	4563	10.1	no	static	medium	sunny	new
Total testing			**5783 (10:07)**	**721**	**61440**	**10.6**					

Figure 2.6: Benchmark dataset-multiple object tracking challenge 2015

Later in 2016 and 2017, the dataset has been modified with more accurate content, including the ground truth labelling and detection results 2.7.

2.1.4 RGBD Princeton Tracking Benchmark Dataset

Recently, the increasing popularity of depth sensors has made it possible to obtain reliable depth easily. This may be a game changer for tracking, since depth can be used to prevent model drift and handle occlusion. Therefore, group from Princeton constructs a



Training sequences										
Name	FPS	Resolution	Length	Tracks	Boxes	Density	Camera	Viewpoint	Conditions	Source
MOT16-02	30	1920x1080	600 (00:20)	49	17,833	29.7	static	medium	cloudy	new
MOT16-04	30	1920x1080	1,050 (00:35)	80	47,557	45.3	static	high	night	new
MOT16-05	14	640x480	837 (01:00)	124	6,818	8.1	moving	medium	sunny	[13]
MOT16-09	30	1920x1080	525 (00:18)	25	5,257	10.0	static	low	indoor	new
MOT16-10	30	1920x1080	654 (00:22)	54	12,318	18.8	moving	medium	night	new
MOT16-11	30	1920x1080	900 (00:30)	67	9,174	10.2	moving	medium	indoor	new
MOT16-13	25	1920x1080	750 (00:30)	68	11,450	15.3	moving	high	sunny	new
Total training			**5,316 (03:35)**	**512**	**110,407**	**20.8**				

Testing sequences										
Name	FPS	Resolution	Length	Tracks	Boxes	Density	Camera	Viewpoint	Conditions	Source
MOT16-01	30	1920x1080	450 (00:15)	23	6,395	14.2	static	medium	cloudy	new
MOT16-03	30	1920x1080	1,500 (00:50)	148	104,556	69.7	static	high	night	new
MOT16-06	14	640x480	1,194 (01:25)	217	11,538	9.7	moving	medium	sunny	[13]
MOT16-07	30	1920x1080	500 (00:17)	55	16,322	32.6	moving	medium	shadow	new
MOT16-08	30	1920x1080	625 (00:21)	63	16,737	26.8	static	medium	sunny	new
MOT16-12	30	1920x1080	900 (00:30)	94	8,295	9.2	moving	medium	indoor	new
MOT16-14	25	1920x1080	750 (00:30)	230	18,483	24.6	moving	high	sunny	new
Total testing			**5,919 (04:08)**	**830**	**182,326**	**30.8**				

Figure 2.7: Benchmark dataset-multiple object tracking challenge 2016/2017

unified benchmark dataset of 100 RGBD videos with high diversity [95], which can be evaluated with either 2D or RGBD/3D tracking algorithms. In Figure 2.8, we can see depth provides valuable information to predict and track the target.

Within all the datasets we have reviewed, the most commonly used one by researchers is still object tracking benchmark dataset (OTB50/100) considering of its diversity and regularity. The surveillance videos are too limited on the targets and the RGBD sequences are still not so popular right now. Therefore, in our work, we focus on OTB dataset and compare the performance with other tracking algorithms.

2.2 Evaluation Methodology

- **Precision plot.** One widely used evaluation metric on tracking precision is the center location error, which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. Then the average center location error over all the frames of one sequences is used to summarize the overall performance for that sequence. If we set a score

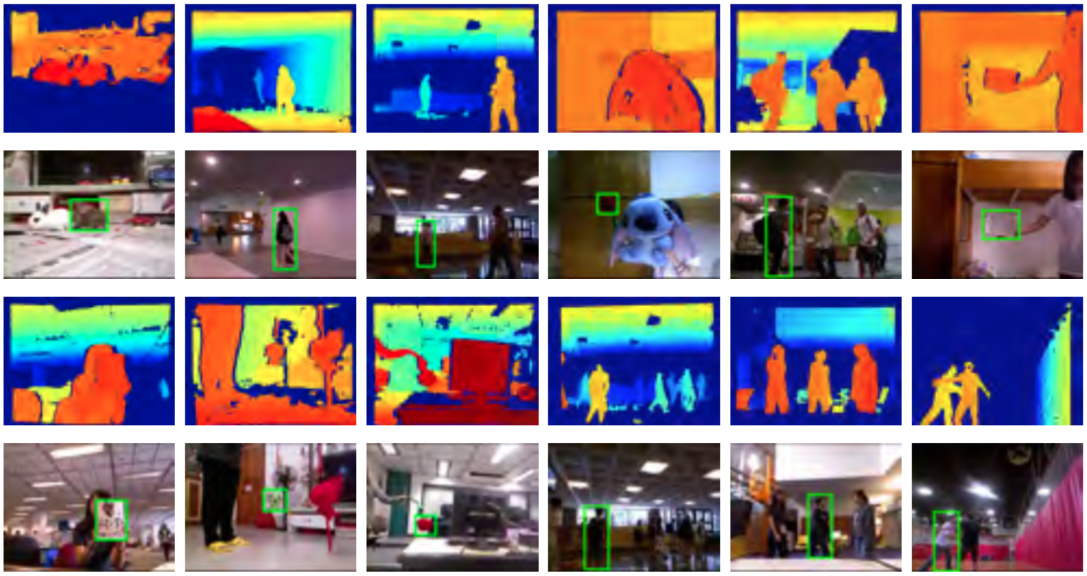


Figure 2.8: Examples from Princeton RGBD tracking benchmark dataset.

(=20 pixels) as the threshold, it shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth.

- **Success plot.** Another evaluation metric is the bounding box overlap. Given the tracked bounding box b_t and the ground truth bounding box b_g, the overlap score is defined as

$$S = \frac{|b_t \cap b_g|}{|b_t \cup b_g|}$$

where $\cap$ and $\cup$ represent the intersection and union of two regions, respectively, and $|\cdot|$ denotes the number of pixels in the region. To measure the performance on a sequence of frames, we count the number of successful frames whose overlap S is larger than the given threshold. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1.

- **Robustness evaluation.** The traditional way to evaluate the performance of a tracker is to initial the system at the first frame with ground truth target location.

This is so called one-pass evaluation (**OPE**). There are also two approaches to measure the robustness of the algorithm with respect to different initializations. One it temporal robustness evaluation (**TRE**) and another is spatial robustness evaluation (**SRE**). In TRE, we do not start tracking from the first frame, but from any middle time spot. In SRE, even though we start tracking from the first frame, the initialization bounding box is not exact the ground truth location but with some shiftings or scalings.

2.3 Important Tools and Techniques in Object Tracking

2.3.1 Optical Flow

The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t + \Delta t$ at every voxel position. These methods are based on local Taylor series approximations of the image signal and use partial derivatives with respect to the spatial and temporal coordinates.

For a 2D+t dimensional case, a voxel at location (x, y, t) with intensity $I(x, y, t)$ will have moved by $\Delta x, \Delta y$ and Δt between the two image frames, and the following brightness constancy constraint can be given:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

Assuming the movement to be small, we have:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T$$

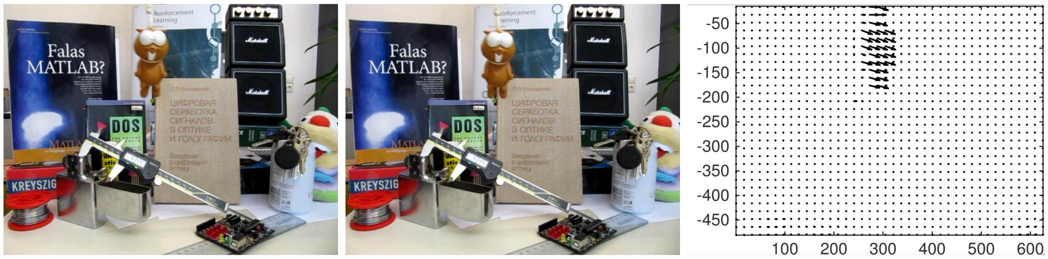


Figure 2.9: Example to illustrate optical flow map.

From these equations it follows that:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0$$

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0$$

where V_x, V_y are the x and y components of the velocity or optical flow of $I(x, y, t)$ and $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the derivatives of the image at (x, y, t).

There are many different ways to solve these equations, such as Lucas-Kanade method, Horn-Schunck method, Black-Jepson method, etc. Figure 2.9 shows one example [98] of what optical flow map looks like. Based on two consecutive frames (left, middle), the pixel-wise motion vector map between them is calculated and shown in the right figure. We can see that this optical flow map provides the temporal motion information and the segmentation cue for the target.

2.3.2 Ensemble Tracking

Ensemble tracking [4] is one of the earliest work that considers tracking as a binary classification problem, where an ensemble of weak classifiers is trained online to distinguish between the object and the background. The ensemble of weak classifiers is combined into a strong classifier using AdaBoost. The strong classifier is then used to label pixels in the next frame as either belonging to the object or the background, giving

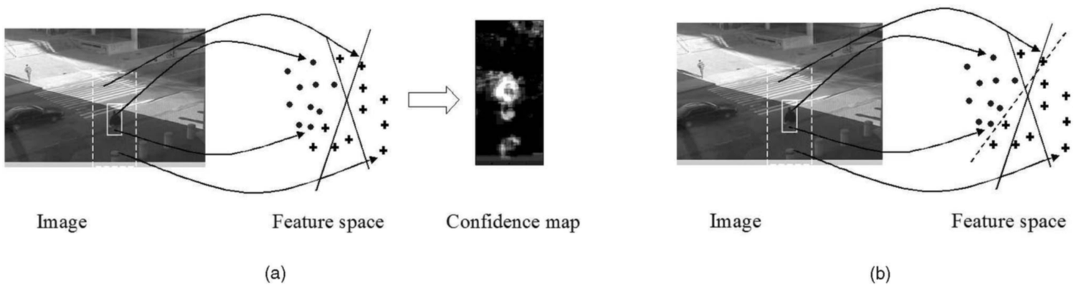


Figure 2.10: Ensemble update and test.

a confidence map. The peak of the map and, hence, the new position of the object, is found using mean shift. Temporal coherence is maintained by updating the ensemble with new weak classifiers that are trained online during tracking.

In Figure 2.10, (a) The pixels of image at time $t - 1$ are mapped to a feature space (circles for positive examples and crosses for negative examples). Pixels within the solid rectangle are assumed to belong to the object, pixels outside the solid rectangle and within the dashed rectangle are assumed to belong to the background. The examples are classified by the current ensemble of weak classifiers (denoted by the two separating hyperplanes). The ensemble output is used to produce a confidence map that is fed to the mean shift algorithm. (b) Now, train a new weak classifier (the dashed line) on the pixels of the image at time t and add it to the ensemble.

2.3.3 Correlation Filtering

Correlation filters have been widely used in numerous applications such as object detection and recognition. Since the operator is readily transferred into the Fourier domain as element-wise multiplication, correlation filters have attracted considerable attention recently to visual tracking due to its computational efficiency.

A typical tracker based on correlation filters models [39, 72] the appearance of a target object using a filter $\mathbf{w}$ trained on an image patch $\mathbf{x}$ of $M \times N$ pixels, where all the circular shifts of $\mathbf{x_{m,n}}$, $(\mathbf{m}, \mathbf{n})$, are generated as training samples with Gaussian function label $y(m, n)$, i.e.,

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{m,n} |\phi(\mathbf{x}_{m,n}) \cdot \mathbf{w} - y(m,n)|^2 + \lambda |\mathbf{w}|^2$$

where ϕ denotes the mapping to a kernel space and λ is a regularization parameter. Since the label $y(m, n)$ is not binary, the learned filter $\mathbf{w}$ contains the coefficients of a Gaussian ridge regression rather than a binary classifier. Using the fast Fourier transformation (FFT) to compute the correlation, this objective function is minimized as $\mathbf{w} = \sum_{m,n} \mathbf{a}(m, n)\phi(\mathbf{x}_{m,n})$, and the coefficient $\mathbf{a}$ is defined by

$$A = \mathcal{F}(\mathbf{a}) = \frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\phi(\mathbf{x}) \cdot \phi(\mathbf{x})) + \lambda}$$

where F denotes the discrete Fourier operator. The tracking task is carried out on an image patch $\mathbf{z}$ in the new frame with the search window size $M \times N$ by computing the response map as

$$\hat{\mathbf{y}} = \mathcal{F}^{-1}(A \odot \mathcal{F}(\phi(\mathbf{z}) \cdot \phi(\hat{\mathbf{x}})))$$

where $\hat{\mathbf{x}}$ denotes the learned target appearance model and $\odot$ is the Hadamard product. Therefore, the new position of target is detected by searching for the location of the maximal value of $\hat{\mathbf{y}}$.

2.3.4 Sparsity-based Collaborative Model

In previous parts, we have seen that optical flow and ensemble tracking focus on determining whether a particular pixel (in the search window) belongs to the target or not,

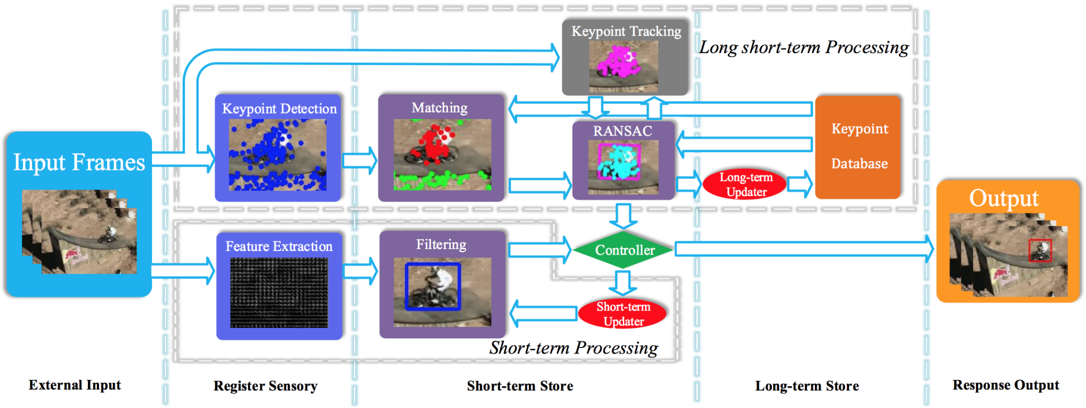


Figure 2.11: A system flowchart of the proposed MUSTer tracker based on the Atkinson-Shiffrin Memory Model.

which refers to a generative model based on local features. While correlation filtering pays attention on the whole target patch, which refers to a discriminative classifier using holistic templates. In order to exploit the strength of both schemes, [129] proposed a sparsity-based collaborative model. It consists of two modules:

- Sparsity-based discriminative classifier (SDC). In this module, the positive and negative patch templates are extracted around the target location to train a classifier for object tracking. In each frame, N candidates will be drawn around the tracked result in the previous frame with a particle filter. Then the classifier will determine whether the candidate is a target or not.

- Sparsity-based generative model (SGM). For generative model, the feature they use is local histogram. They form a local histogram feature vector of the searching window and determine where the target is.

2.3.5 MUlti-Store Tracker (MUSTer)

Inspired by the well-known Atkinson-Shiffrin memory model, author [43] proposed a MUlti-Store Tracker (MUSTer), a dual-component approach consisting of short- and long-term memory stores to process target appearance memories.

In Figure 2.11, we can see that the short-term processing unit contains two parts: a correlation filtering tracker based on HoG feature and a SIFT keypoint matching/tracking with RANSAC estimation. For long-term processing unit, they build a keypoint database to store the historical target information. Both the results of short-term processing and long short-term processing are obtained by a controller, which decides the final tracking output.

2.3.6 Visual Tracking using Convolutional Features

Recently with the development of convolutional neural network in computer vision tasks, researchers try to use convolutional features in visual tracking [71, 109]. They actually did not train a CNN tracking model, but exploited features extracted from VGGNet, which trained on object recognition datasets, to improve tracking accuracy and robustness.

Work in [71] shows that early CONV layers preserve more spatial content pattern of the object and FC layers contain more semantics information, as shown in Figure 2.12. Based on rich hierarchical features of CNNs, the system adaptively learn linear correlation filters on each CNN layer to alleviate the sampling ambiguity and infer the target location using the multi-level correlation response maps in a coarse-to-fine fashion. Figure 2.13 shows the main steps of their proposed tracker.

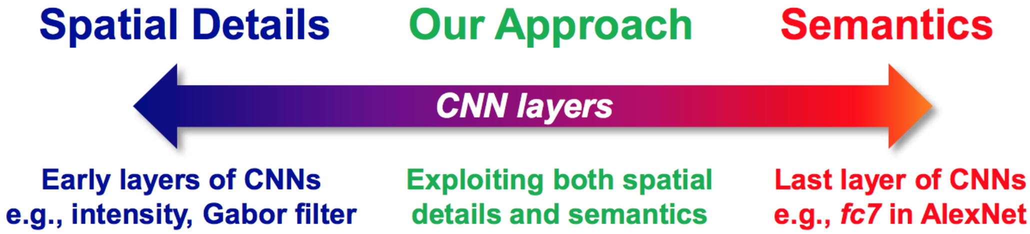


Figure 2.12: Convolutional layers of a typical CNN model.

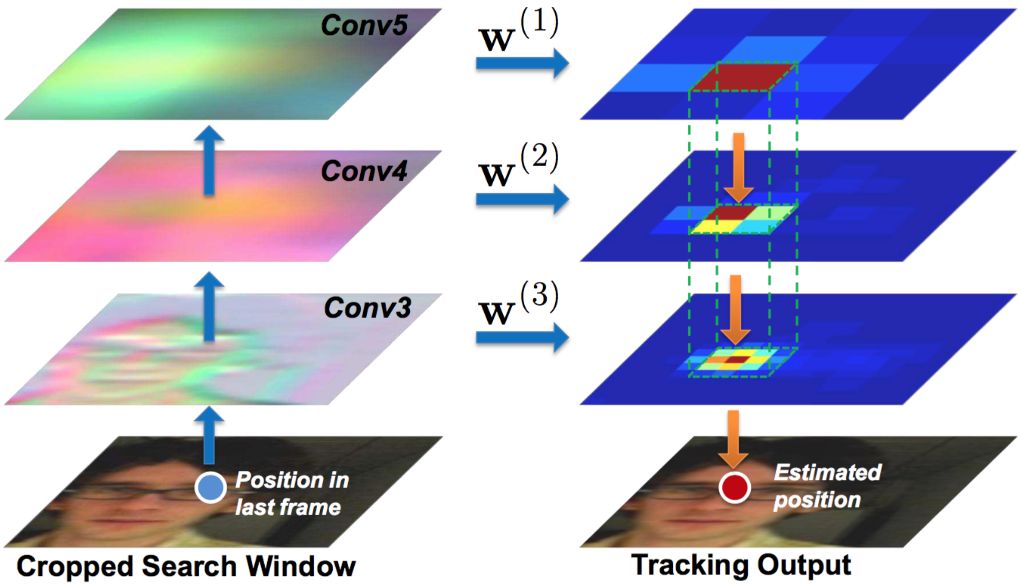


Figure 2.13: Main steps of the proposed algorithm. Given an image, we first crop the search window centered at the estimated position in the previous frame. We use the third, fourth and fifth convolutional layers as our target representations. Each layer is then convolved with the learned linear correlation filter to generate a response map, whose location of the maximum value indicates the estimated target position. We search the multi-level response maps to infer the target location in a coarse-to-fine fashion.

Chapter 3

Object Tracking with Temporal Prediction and Spatial Refinement (TPSR)

3.1 Introduction

Most of the state-of-the-art online object tracking algorithms adopt a model-based approach with some features and/or representation. The models include correlation filters, Markov Chain Monte Carlo, graph models, the conditional random field, etc. Significant progress has been made with this parametric approach in recent years. However, the tracking performance of these algorithms appears to reach a plateau due to multiple challenging attributes such as illumination variation, occlusion, deformation, fast motion and background clutters, etc. Besides, there exist apparent visual tracking errors which are not properly reflected by performance indices. In general, the model-based approach has two major limitations. It is difficult to develop a robust model to deal with a wide range of video contents. It is also difficult to identify real causes of errors for further improvement.

As compared with state-of-the-art methods in the literature, a non-parametric online object tracking system based on the idea of temporal prediction and spatial refinement (TPSR) is proposed in this chapter. The proposed tracking system consists of three cascaded modules: pre-processing (PP), temporal prediction (TP) and spatial refinement

(SR) of the tracking bounding box. Illumination variation and camera shaking are compensated in the PP module. A joint template matching (TM) and optical flow (OF) scheme is adopted in the TP module, where TM and OF often complement each other. For example, TM fails when an object and its background are with textured surfaces since there are many local minima in the matching results. OF provides a pixel-wise motion field, and it is particularly effective in determining the movement of textured surfaces. Finally, the drifting problem is handled in the SR module. To control error accumulation, the location and size of the predicted bounding box from the previous module are finetuned using the local spatial information of the new frame. It is a critical module since the object-environment relationship frequently changes in the new frame due to camera zoom-in/out, rotational motion and shape deformation, etc. This module is extremely valuable in avoiding tracking error accumulation over multiple consecutive frames.

There are two major contributions in this work. First, the use of TM and OF jointly for robust temporal prediction of the bounding box between the reference and the current frame in the TP module is new. Second, the refinement of bounding box's location and size using the spatial domain information of the new frame in the SR stage is also novel. It offers an effective mean to control cumulative errors due to temporal prediction alone. The two ideas, TP and SR, enhance the online object tracking performance substantially.

The rest of this chpater is organized as follows. An overview of the proposed TPSR tracking system is presented in Section 3.2. Its tracking performance is shown and compared with a few state-of-the-art methods against a benchmarking dataset in Section 3.3. Finally, concluding remarks are given in Section 3.4.

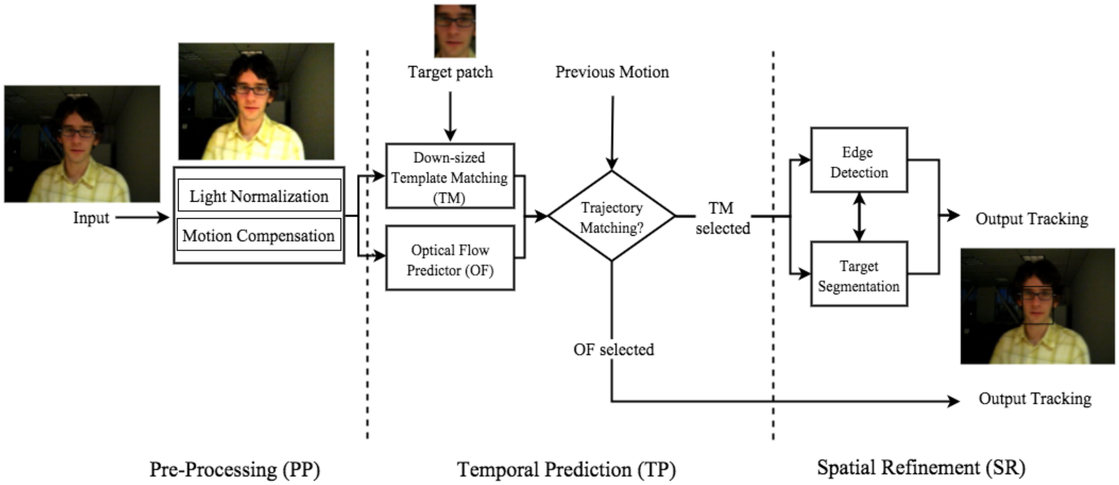


Figure 3.1: The flow chart of the proposed TPSR online object tracking system.

3.2 Proposed TPSR Tracking System

3.2.1 System Overview

As shown in Figure 3.1, the TPSR method consists of the following three cascaded modules.

1. **Pre-Processing (PP).** Light normalization and camera motion compensation are conducted to reduce the impact of illumination variation and bounding box orientation mismatching between two consecutive frames due to camera shaking.

2. **Temporal Prediction (TP).** A joint TM/OF scheme is used to establish the correspondence of the object region between two consecutive frames. The complementary characteristics of TM and OF enable robust tracking under various object-environment settings.

3. **Spatial Refinement (SR).** Although the correspondence of the object regions between two consecutive frames is built in the TP module, the object-environment relationship may change in the new frame. We refine the bounding box location

and size using the spatial information of the new frame, and use this idea to control tracking error accumulation.

Details of each module are elaborated in the following subsections.

3.2.2 Pre-Processing (PP) Module

Lighting Normalization. Poor or rapidly changing illumination is one of the main challenges in object tracking. There are two scenarios for consideration.

- Dark scenes. Both the object and its environment are dark. It is difficult to find a clear boundary to separate the object from its background, even for human vision.
- Temporal illumination change. The sudden change in the lighting condition (such as scenes with flash light and thunderstorm) will result in rapidly changing pixel values so that both feature-based and template-based matching methods are not accurate.

Lighting normalization is used to address these challenging lighting conditions. For a scene with dim environmental light, rather than equalizing illumination with respect to the reference frame, we enhance the contrast of the object and its surrounding environment by applying histogram equalization to the current frame. Then, the object becomes more visible for extraction. For sudden illumination change, the object is difficult to track even if the light is normalized. Then, we skip the current frame and move to the next frame. The bounding box of the skipped frame will be interpolated based on the reference frame and the new current frame when the global ambient change disappears.

Let $H_F(t)$ be the histogram of a frame at time t. The skip mode is determined by the following condition:

$$skip = \begin{cases} 1 & \text{if } D(H_F(t), H_F(t-1)) > T, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

where T is a threshold and $D()$ is a Chi-squared function used to represent the difference between two histograms. If we decide to skip frame t, the reference frame for frame $t+1$ is still frame $t-1$. The purpose of frame skipping is to ensure that the object in the reference and the current frames share a similar illumination condition.

Camera Motion Compensation. There is irregular camera motion due to hand-held camera shooting in some video sequences. It will confuse the following tracking modules greatly. Thus, camera motion compensation is essential to a robust tracking system. There exist numerous techniques to stabilize the scene. In the PP module, camera motion estimation is estimated by finding a perspective transformation between two frames with respect to the background scene. In our implementation, we extract the SIFT descriptor [70] and then find the transform matrix based on RANSAC matching. The obtained transform matrix aligns the views of two consecutive frames, and the compensated frame serves as the new input to the following tracking modules. Once the tracked object is determined, it will be transformed back to the original frame to be displayed as the final tracking result.

3.2.3 Temporal Prediction (TP) Module

The temporal prediction (or the tracking procedure) component of most state-of-the-art single object tracking (SOT) methods follows the tracking-by-detection (TBD) framework. It adopts a discriminative model that separates the target region from the background in the search window. It keeps updating while maintaining the long/short term memory of the target along time. Since the TBD framework treats the tracking problem as a spatial detection problem, it is difficult to single out a temporal prediction component from these methods. This is the reason we do not elaborate on the temporal prediction of existing methods. On the other hand, our TPSR tracking system does contain temporal prediction as one building component, where template matching (TM) and optical flow (OF) predictor work jointly for target localization. We allow these two modes to work simultaneously and leverage their complementary strength.

This idea enables the TPSR system to track a target along its trajectory accurately under various complex situations. TM offers a displacement vector of the bounding box between the reference and the current frames based on region similarity. OF provides a pixel-wise motion field, where a region with a consistent motion field determines the displacement vector. TM and OF can complement each other in various situations. For example, TM fails when an object and its background are both with textured surfaces while OF is effective in determining the movement of textured surfaces. On the other hand, OF may fail for an object with a large homogenous region while TM can provide a good result if there is a reasonable boundary between the object and its environment. Besides, the computational complexity of OF is higher.

TM Motion Predictor. Tracking an object in the pixel domain of the original image is neither efficient nor robust due to its high pixel dimension and variation of surface details. To address this issue, we adopt a downsized bounding box as the matching template. The original bounding box may be of various sizes and in rectangular shape.

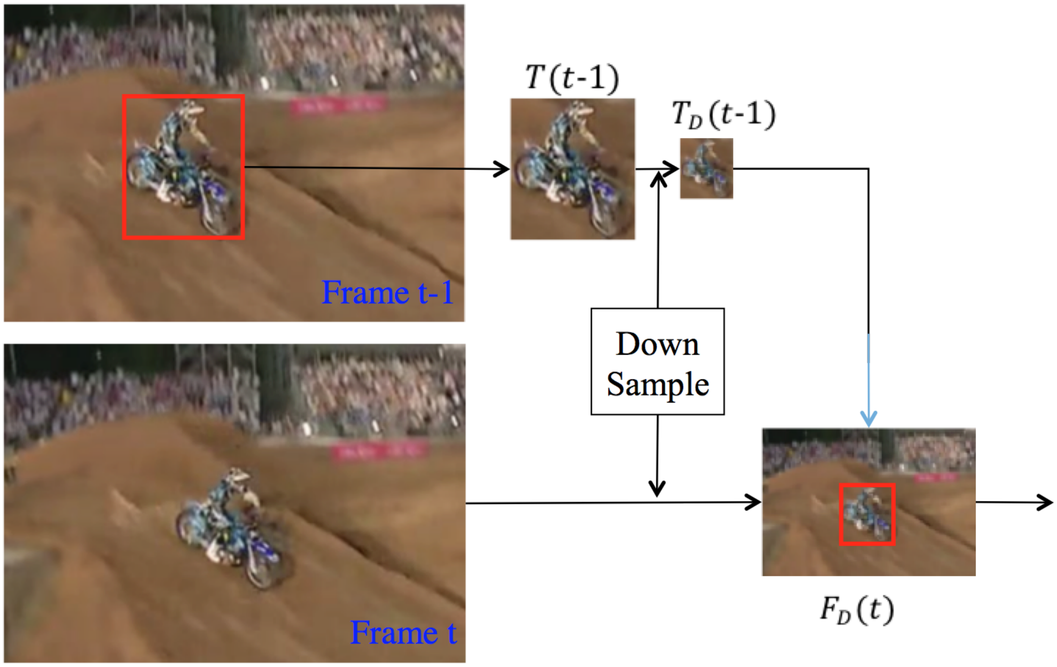
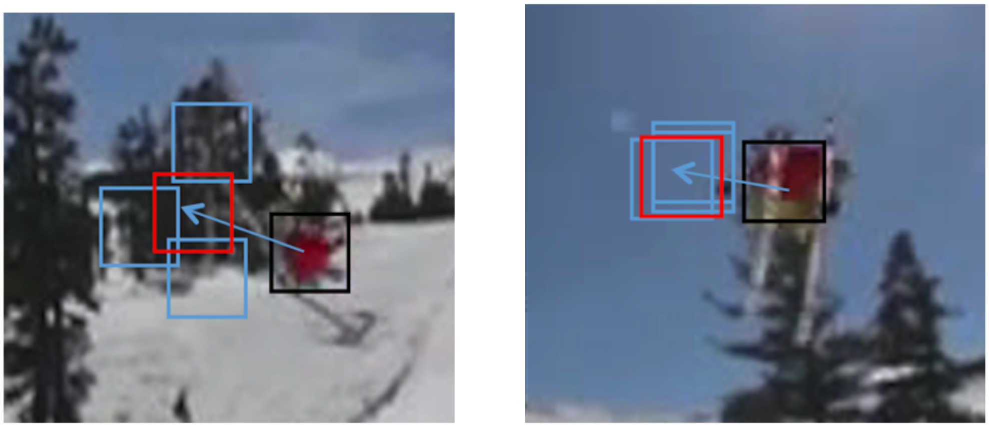


Figure 3.2: The use of a downsized object region $T_D(t-1)$ in the reference frame to search for the best match in the downsized current frame $F_D(t)$.

Here, we set the shorter dimension of a downsized bounding box to be 16 pixels. After that, the image within the search window is also downsized proportionally. The purpose of downsizing is to remove unnecessary spatial details so as to make the performance of TM more robust. An example of template down-sizing is shown in Figure 3.2. The red box, $T(t-1)$, in the figure represents the object region detected in frame $t-1$, while its downsized version, $T_D(t-1)$, serves as the initial position of prediction in downsized frame t. Furthermore, we give different weights to matching errors of pixels within the bounding box. That is, we assign more weights to the object region than the background and choose the location of the best match with the minimum sum of weighted absolute difference (SWAD) in the Lab color space as the TM prediction result.

OF Motion Predictor. When an object moves in front of cluttered background, the TM motion predictor tends to give a poor prediction. Then, the OF motion predictor


(a) TM to OF

(b) OF to TM

Figure 3.3: System switch between TM and OF. Black rectangle indicates the previous location. Red one is the trajectory prediction. Blue ones represent the TM candidates. (a) TM candidates do not match trajectory predication, the proposed system switches to OF. (b) TM candidates matches the trajectory prediction, the proposed system switches back from OF to TM.

[98] can be adopted as an alternative to acquire the pixel-wise motion of the object. OF provides a pixel-wise motion field, where a region with a consistent motion field determines the displacement vector.

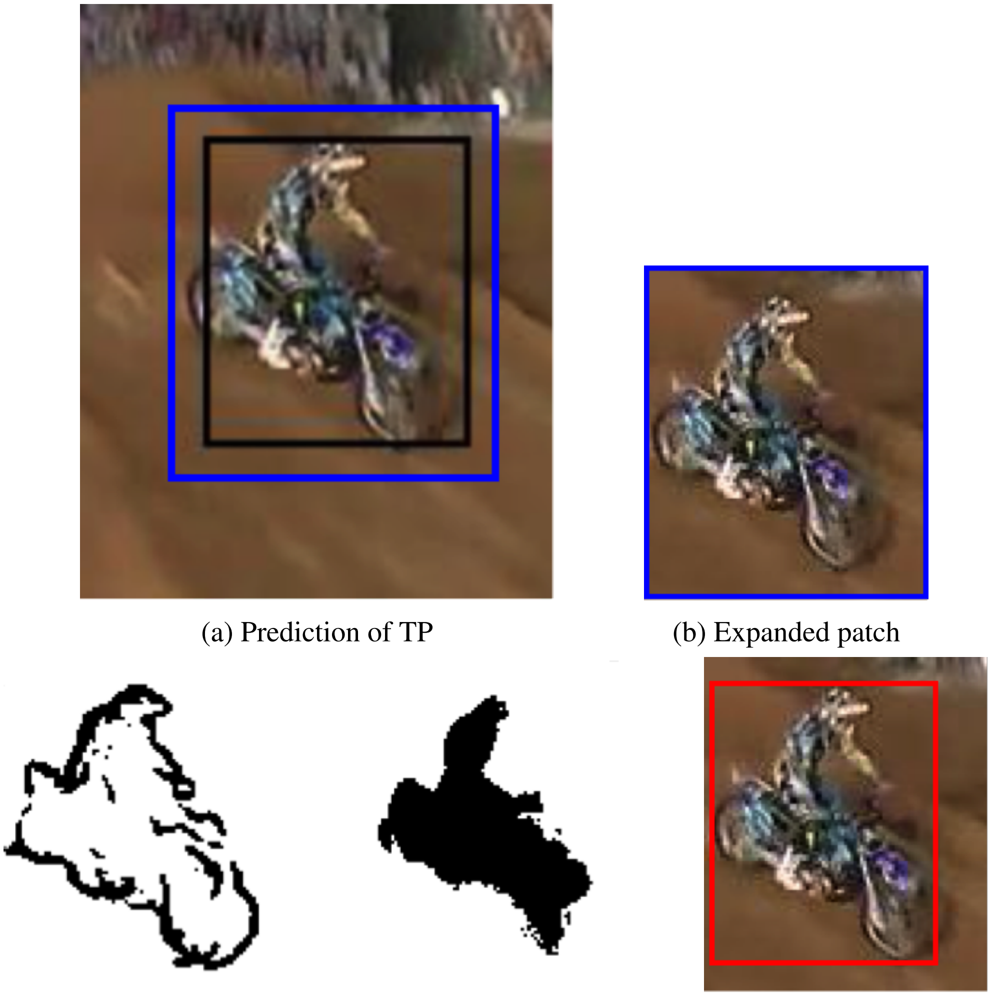
Automatic TM/OF Switching. In the beginning of each sequence, we adopt both TM and OF motion predictors and compare their tracking performance to decide which motion model works better. Then, we stick to one motion model in the sebsequent frames until a model switching criterion is met. This criterion is determined by two factors. First, the predicted bounding box location deviates much from the motion trajectory of the moving object across multiple frames in the past. Second, the alternative predictor provides a result that is more consistent with the motion trajectory. This idea is shown in Figure 3.3.

First, we consider to switch from TM to OF. Three locations predicted by TM with the smallest SWAD values are considered as prediction candidates. The motion trajectory can be easily computed based on the path of the bounding box center. The TM motion predictor chooses the candidate that is closest to the predicted trajectory as the final prediction. The prediction result is satisfactory if the bounding box of the reference frame has a sufficiently large overlap with that of the current frame. Otherwise, the TM predicted result is not reliable, then it will turn on the OF motion predictor. Next, we consider to switch from OF to TM. The strategy is similar to the first one but reverse the role of TM and OF. The OF motion predictor will predict only one new location instead of three. If the predicted location deviates much from the motion trajectory, we will compute the three locations predicted by TM with the smallest SWAD values and choose the one that is most consistent with the motion trajectory.

3.2.4 Spatial Refinement (SR) Module

The TP module provides the best predicted bounding box position based on the comparison of two consecutive frames (the reference and the current frame). However, a good match does not imply an accurate object location in the current frame. There are several reasons. First, template matching (TM) is based on the similarity matching for downscaled target patch. The down sampling operation does not guarantee a pixel-wised accuracy. Second, for optical flow prediction (OF), we find the average motion vectors of all the target pixels. But the accuracy is not promised because of the part deformation or rotation, which will not generate a correct target movement. Third, the object-environment relationship may change between consecutive frames. For example, an object moves from clean to cluttered background, or has rotational motion and/or shape deformation, etc. When there is a prediction error, it tends to accumulate along time. As a result, the object of interest will drift away from the tracking bounding box. There is no mechanism to compensate for drifting errors in traditional online object tracking methods.

In the proposed TPSR method, we use the SR module to control the drifting error based on the spatial information of the current frame. We use the structure edge detector [25] and the GrabCut segmentation algorithm [67] to achieve this goal. The input to these two algorithms is an expanded version of the bounding box obtained in TP as illustrated in Figure 3.4(b). The reason of using the expanded version is that the bounding box obtained in TP may lose some object boundaries due to the prediction error. Simple edge detection tools such as the Sobel edge detector and the Canny edge detector are too sensitive to noise and do not meet the purpose. As shown in Figure3.4(c), the structure edge can provide a more accurate object boundary. Another tool to separate the object from its background is object segmentation where the segment number is equal to two. The GrabCut segmentation algorithm is a powerful tool that needs seeds for the object and background. The information provided by the structure edge can provide seeds within the object. The seeds of background can be sampled from boundaries of the expanded bounding box region. The result of GrabCut is shown in Figure 3.4(d). Finally, based on the boundary and segmentation results, an accurate bounding box location can be found in Figure3.4(e).



(a) Prediction of TP

(b) Expanded patch

(c) Structure edge

(d) GrabCut

(e) Final location

Figure 3.4: SR Module: (a) Prediction result (black) from TP module, which contains some potential issues, such as boundary alignment. (b) Expanded patch of the prediction result (blue). (c) Structure Edge result of the expanded patch. (d) GrabCut segmentation of the expanded patch. Based on the results in (c) and (d), more accurate target boundary can be found in red(e).

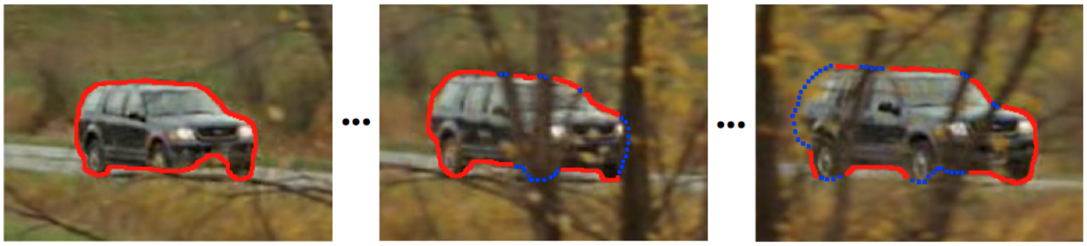


Figure 3.5: Occlusion handling: red and blue segments indicate visible contour and inferred contour, respectively.

3.2.5 Occluded Object Tracking

It is a major challenge to any tracking method when the tracked object is occluded partially or fully. We incorporate a salient contour tracking scheme in the spatial refinement module to address this problem. The scheme is based on one underlying assumption - the object be tracked should have one or multiple salient contours. Examples include the outer boundary of an object, the eye glass of a human, etc. Usually, the full view of the object is available in the beginning of the sequence and we can extract salient contours of the object. Although the contours may deform along time and could be blocked by occluders, they should not disappear suddenly. As a result, if we do not see these contours in the current frame, we have to infer their location based on the prior in the reference frame.

This idea is illustrated by an example in Figure 3.5, where the car is the object to be tracked. In the first several frames, the car has a clear outer contour which will be extracted in the spatial refinement module. However, when the car is occluded in the middle set of frames, its outer contour becomes confusing due to occlusion. However, we have to make an inference on the counter location based on its prior in the reference frame. In Figure 3.5, we use the red color to denote the visible contour segment and the blue color to denote the inferred contour segment. Although these two segments are labeled by different colors, they form a complete contour jointly. Then, it will serve as the prior for the next frame. If the object is fully occluded, none of its contour is visible. However, if it is still being tracked, we have its whole contour in blue, indicating a completely inferred contour. Although there is no temporal prediction and spatial refinement in the full occlusion case, we assume that the object keeps moving along its motion trajectory at the same speed without any shape variation. This assumption often holds if the full occluding time is short.

3.3 Experimental Results

3.3.1 Dataset

The performance of the TPSR method is benchmarked with quite a few state-of-the-art algorithms against a well known dataset in [112] with one pass evaluation (OPE). This dataset contains 50 video sequences with various challenging attributes such as fast motion, illumination variation, deformation, etc. The evaluation was conducted on an 2.3 GHz Intel Core i7 CPU with 16 GB RAM. The current implementation has not been optimized and its processing rate is 0.4 fps. The processing speed can be easily improved after software optimization. Furthermore, quite a few operations such as template matching and optical flow computation can be done in parallel. Their speed can be even further improved by suitable supporting hardware (e.g., GPU or multi-core CPU).

We compared more than 20 state-of-the-art methods in this work. Since all of them were run on the same benchmarking dataset known as the Visual Tracker Benchmark (VTB) library [112], we adopted their reported results. For example, we downloaded pre-computed tracking results of the MEEM tracker [124] from authors' webpage. We also got the source code and the reported results of the LCT tracker [72] from authors'

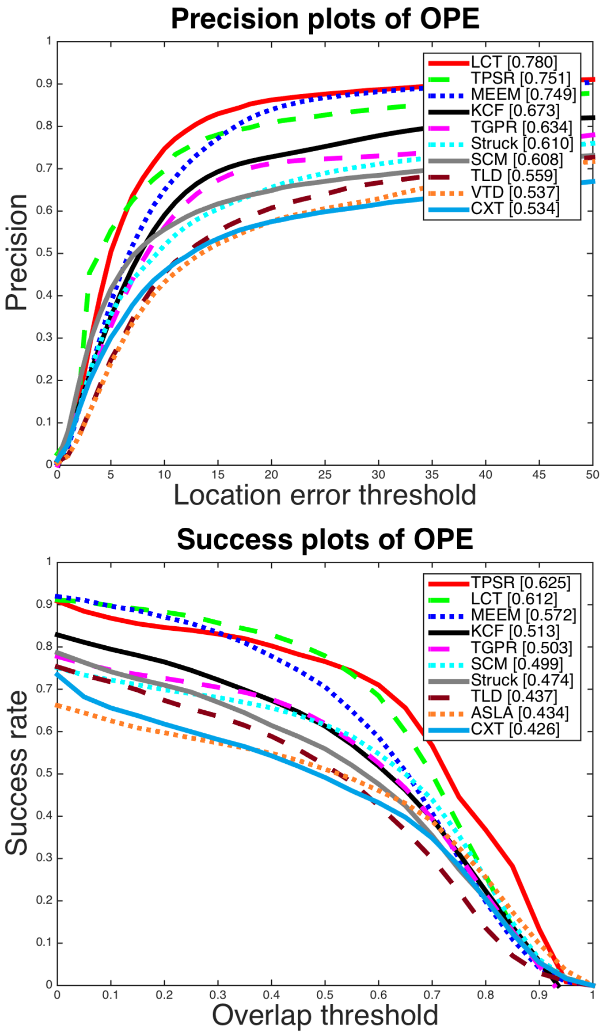


Figure 3.6: The location precision and success rate performance curves of top 10 tracking algorithms using 50 video sequences. The AUC value of the TPSR method ranks No. 1 in the success rate plot and No. 2 in the location precision plot.

webpage. To make fair comparison, we use the default parameter setting given by the benchmark library for our proposed algorithm.

3.3.2 Overall Performance

Given two bounding boxes (i.e., one of the ground truth and the other of a tracking algorithm), it is typical to evaluate the tracking algorithm with two metrics: the center location error and the overlap success rate. The former is the averaged Euclidean distance between two centers while the latter is the percentages of frames where the overlapping region between two bounding boxes surpasses a threshold. Then, two performance curves can be drawn by varying their threshold values as shown in Figure 3.6. The x-axis of the location error plot is the number of pixels and that of the success rate plot is the overlap region percentage with respect to the bounding box of the ground truth.

We compare the performance of more than 20 methods and show the results of the top ten. They are the proposed TPSR method, MEEM [124], KCF [39], TGPR [33], SCM [129], Struck [38], TLD [49], ASLA [48], CXT [24], VTD [61], ALIEN [86] and LCT [72]. The area under the curve (AUC) values of the TPSR method are 75.1% and 62.5%, which rank No. 1 and No. 2, for the location precision plot and the success rate plot, respectively. We see that TPSR and LCT are two close competitors whose ranks are reversed in these two plots.

Furthermore, the temporal robustness evaluation (TRE) and the spatial robustness (SRE) evaluation are conducted. We show the overlap success plots for TRE and SRE in Figure 3.7. The proposed tracker performs well against other state-of-the-art methods. Based on the successful integration of temporal prediction (TP) and spatial refinement (SR), the proposed TPSR tracker can stabilize the target location under different initializations.

The TPSR method can track the object well with a stricter requirement on the overlap region. It outperforms other methods by a significant margin when the threshold is chosen to be 60% or higher. It means that the TPSR method can predict the location

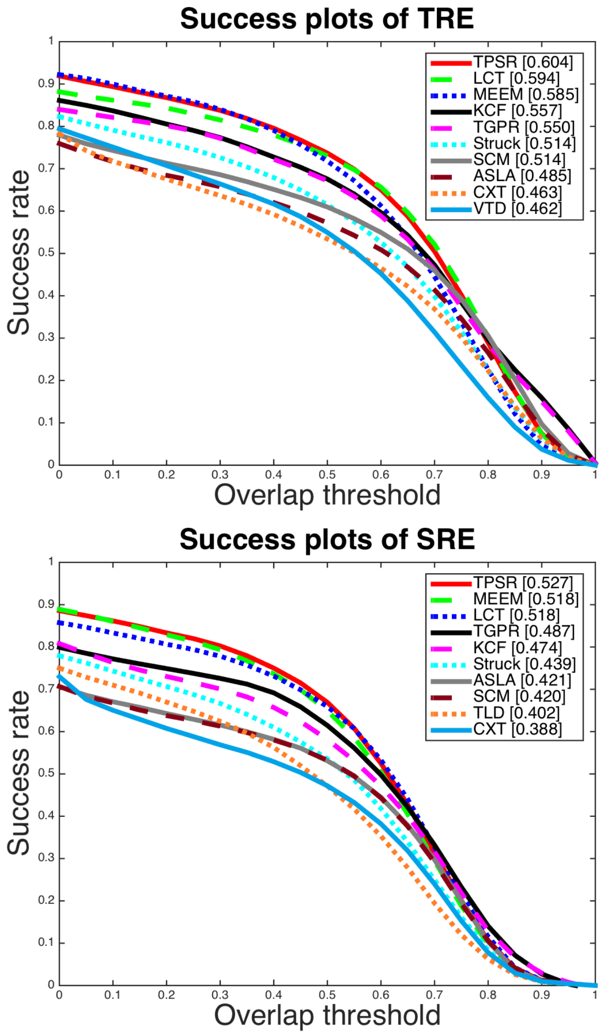


Figure 3.7: Performance comparison in terms of temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE), where the overlap success plots for TRE and SRE with respect to 50 benchmark sequences are shown in the figure.

and size of the object well. As to the location error, the TPSR method offers the best performance when the location error is less than or equal to 7 pixels. It becomes the second and the third best when location errors are between 7 and 14 pixels and higher than 14 pixels. It is worthwhile to comment that the absolute location errors may not be as informative as the relative location errors where errors along the x-axis and the

Sequence	ALIEN	TPSR
David	98	96
Jumping	87	96
Pedestrian1	100	99
Car	100	99
Girl	66	92
Sylvester	98	96
FaceOcc1	99	100
FaceOcc2	100	100
Tiger	30	98
Lemming	38	99
Liquor	81	93
Trellis	92	90

Table 3.1: Comparison of success rates (%) between the ALIEN and the TPSR trackers.

y-axis are normalized by the width and the height of the ground truth bounding box, respectively.

We conduct another experimental comparison between the ALIEN tracker [86] and the proposed TPSR tracker in Table 3.1. Since the object bounding box is oriented in the ALIEN tracker, there is a slight penalty of the ALIEN tracker by forcing it to consider the smallest axis-aligned bounding box that contains the object bounding box. Also, based on experimental results in [86], the performance of sequence-by-sequence success overlapping rates are compared in Table 3.1. We see from the table that the proposed TPSR tracker outperforms the ALIEN tracker significantly in five sequences; namely, Jumping, Girl, Tiger, Lemming and Liquor. For the remaining seven sequences, the two trackers have comparable performance.

3.3.3 Attribute-based Evaluation

The benchmarking dataset contains a wide range of video sequences with six main attributes: 1) fast motion, 2) motion blur, 3) deformation, 4) occlusion, 5) scale variation and 6) illumination variation. A video sequence may have multiple attributes. It is common to evaluate the performance of different tracking methods with respect to sequences of different attributes. This study is valuable in understanding the strength and weakness of each tracker in the presence of different challenging conditions. We show the success rate plot for the ten best results in each case in Figure 3.8. As shown in the figure, the TPSR method gives the best AUC performance for video sequences with four attributes; namely, fast motion, motion blur, deformation and scale variation. It ranks the second for the other two attributes - occlusion and illumination. Actually, it is only second to the LCT method by 1.3% and 0.3% in occlusion and illumination, respectively. The robustness of the TPSR method is clearly demonstrated in the attributed-based analysis. The success of the TPSR method can be attributed to the following four factors.

1) Motion Handling. For video sequences of fast motion and motion blur attributes, a tracking-by-detection scheme alone usually does not work well because of unstable feature representations. The TPSR method has two powerful tools to address these challenges. First, it exploits a complementary strategy by integrating the TM and the OF predictors. They do not rely on local features but pixel values in the region. With the help of template down-sizing, the motion blur effect can be reduced to certain degree. With a proper switch between TM and OF, the TPSR method is more robust to different object/scene combinations. Second, the TPSR method does not take the result of the temporal prediction as is. Instead, it adds the spatial refinement module based on the spatial information in the current frame only. It extracts local contours and segments the object from the background. As a result, the TPSR method tends to forget the past and adapt to the new environment faster. This characteristics is very valuable to sequences with fast changing scenes. It also helps avoid tracking error propagation.

2) Appearance Variation Handling. The variation of object appearance imposes another major challenge to the tracking problem. This shows up in form of scale variation and/or deformation. Again, both can be well handled by the TPSR method in the spatial refinement step. The TRSR adopts an expanded window to ensure the object is within the window. Then, with the help of the structural edge detector and the Grabcut segmentation tool, the new object boundary can be determined accordingly without the constraint of the old object boundary. Thus, as long as the TM module of the TPSR tracker can predict a reasonable initial location of the next window, it will work well.

3) Illumination Variation Handling. The TPSR method handles illumination variation through light normalization in the pre-processing module.

4) Occlusion Handling. The TPSR method handles partial/full occlusion through salient contour tracking as described in Sec. 3.2.5.

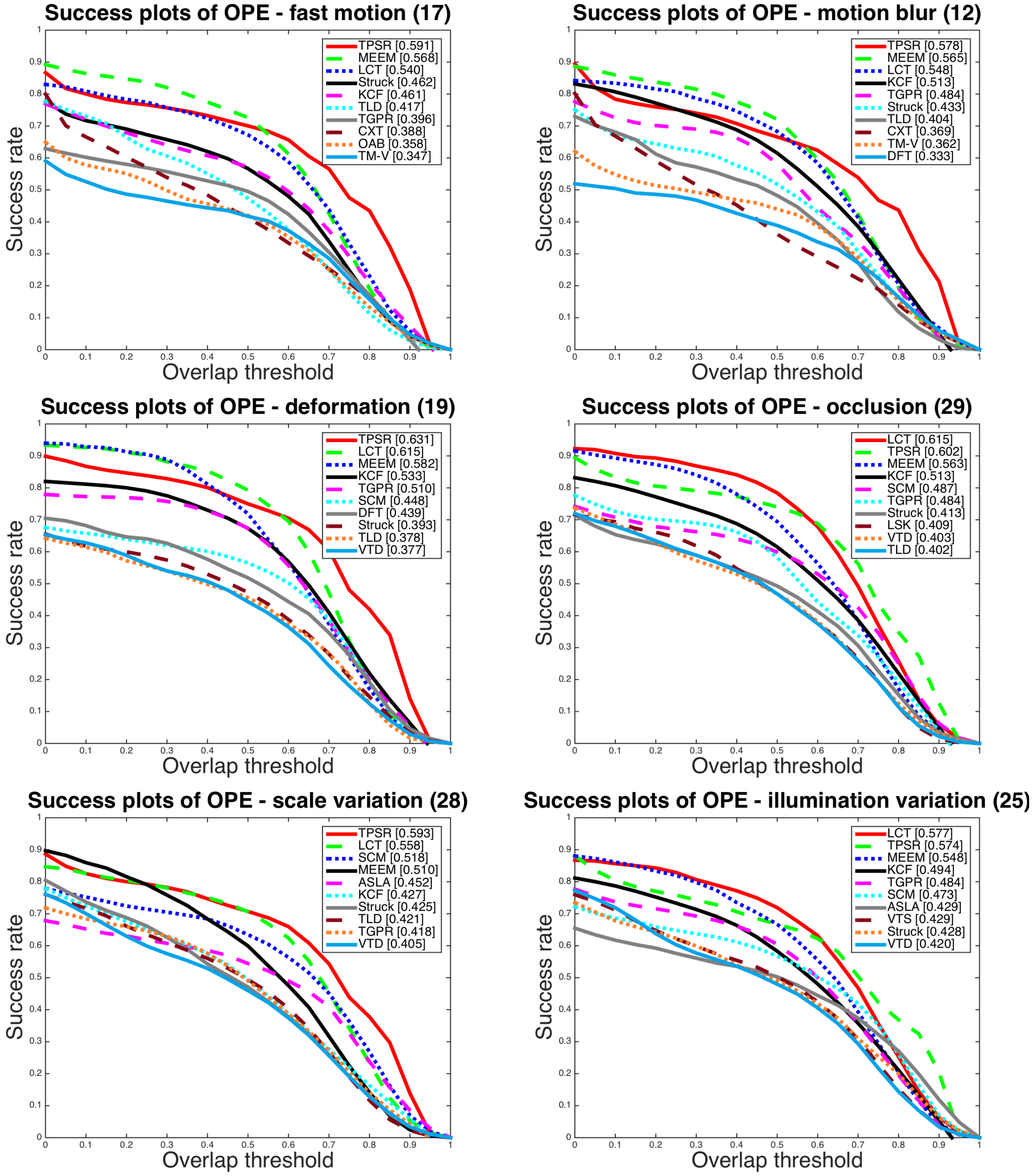


Figure 3.8: Success rate plots with respect to six attributes of video sequences: fast motion, motion blur, deformation, occlusion, scale variation and illumination variation.

3.3.4 Qualitative Evaluation

We select eight representative sequences from the benchmark dataset in [112] and show the tracking results of five algorithms in Figure 3.9 for qualitative evaluation. They are TPSR, LCT, MEEM, Struck and SCM.

The first three sequences in Figure 3.9 are Lemming, Coke and Tiger2. They all have static background and share similar challenging attributes such as background clutter, object deformation, scale variation and occlusion. The TPSR method captures the object location and size well up to their rightmost frames. This is attributed to its powerful OF predictor that offers a good initial location of the object in the current frame against static background. After the OF predictor is selected for a couple of consecutive frames in the beginning stage, it becomes the default one in the tracking process. When occlusion occurs, the system can still track the object based on the occlusion handling technique described in Sec. 3.2.5. Some competitive methods do not perform as well since they are sensitive to the background clutter effect and the erroneous bounding box location and size will propagate into future frames.

The fourth sequence in Figure 3.9 is CarScal. It is challenging because of occlusion and scale variation. We see clearly from the rightmost frame that all methods lose track of the full car except for TPSR due to its powerful spatial refinement module.

The fifth and sixth sequences in Figure 3.9 are Basketball and Skating1. The TM predictor of the TPSR method offers a good initial bounding box for the object. Due to human body's deformation, existing tracking methods such as MEEM, Struck and SCM are not robust enough to track the object for a long while. This is especially true in the presence of occlusion. In contrast, the TPSR method uses a normalized template matching so that it is not sensitive to shape variation. It is also worthwhile to point out that LCT outperforms TPSR at the end of the Skating1 sequence because

LCT considers the long-term correlation of the object shape while TPSR only uses a short-term representation of the object shape.

The seventh sequence in Figure 3.9 is Jumping, which has the effect of motion blur. The TPSR method can successfully track the object based on the TM predictor across all frames. For comparison, we observe that LCT and SCM may lose the object at certain frames although they can recover the object locations at later frames by chance.

The eighth sequence in Figure 3.9 is Skiing. Traditional trackers such as LCT, Struck and SCM lose the object quickly because of background clutter. Only TPSR and MEEM can maintain the correct object location across all frames. This is an example to demonstrate the power of using the TM and the OF predictors in a complementary fashion in handling background clutter under the trajectory constraint. Furthermore, we see from the last two subfigures of the Skiing sequence that TPSR can adjust the bounding box size more flexibly to provide a more accurate size than MEEM. This advantage comes from its spatial refinement module where the object boundary can be re-adjusted based on the spatial information of the current frame only.

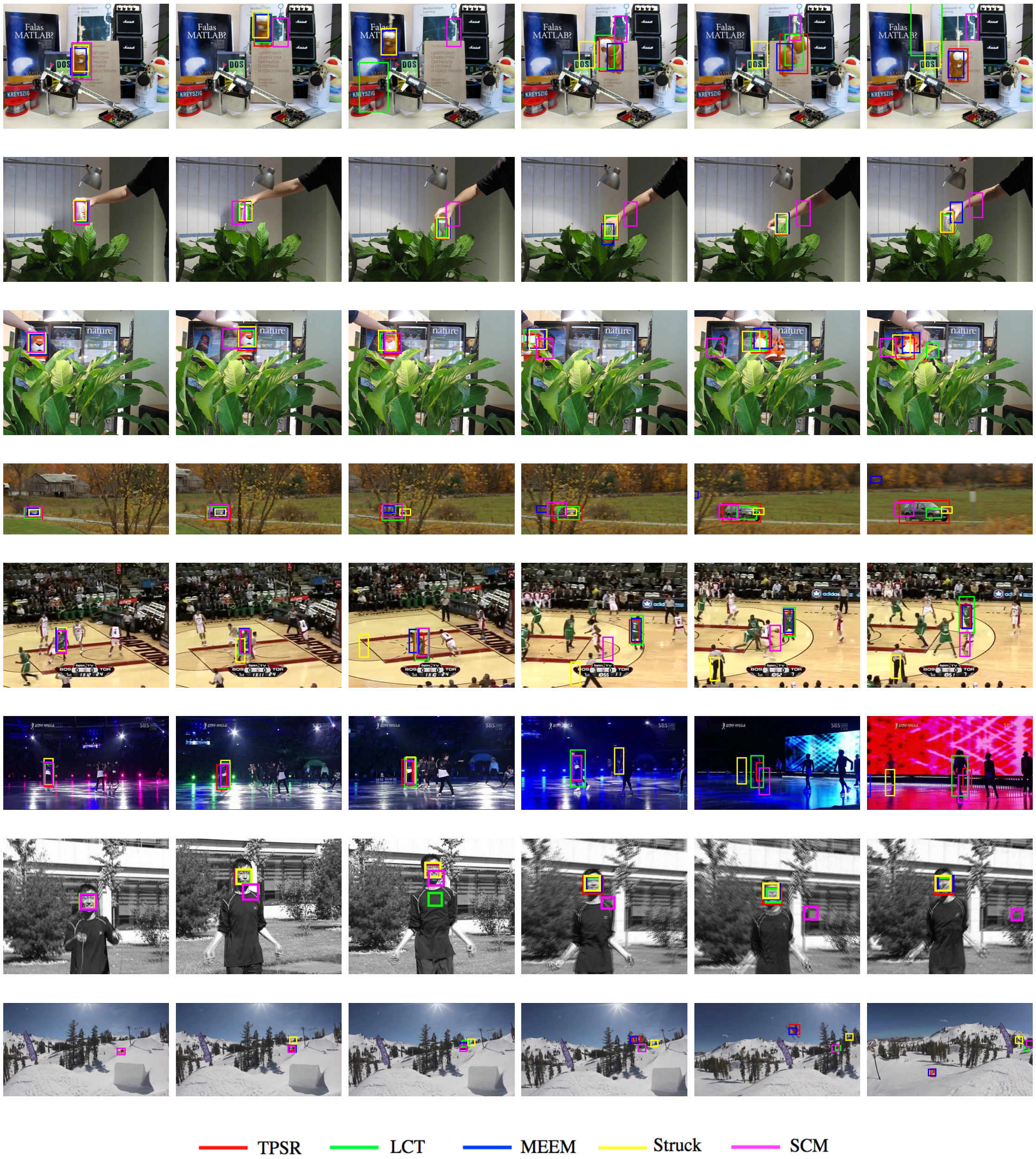


Figure 3.9: Performance visualization of proposed TPSR, LCT, MEEM, Struck and SCM methods on eight challenging sequences (top to down are Lemming, Coke, Tiger2, CarScale, Basketball, Skating1, Jumping and Skiing, respectively).

3.4 Conclusion

A robust online single object tracking (SOT) system based on temporal prediction and spatial refinement, called the TPSR method, was proposed in this work. It has several unique features. First, it uses two temporal predictors (TM and OF) in a complementary fashion. Second, it has a powerful spatial refinement module to make its tracking performance more robust with respect to object shape and size variation. Third, it has a special mechanism to handle either partial and full occlusion. Extensive experimental results were given to demonstrate that the propsed TPSR method offers the state-of-the-art tracking performance in solving the online SOT problem.

Chapter 4

Online Object Tracking via Motion-Guided Convolutional Neural Network (MGNet)

4.1 Introduction

In previous chapter, we describe one traditional solution for visual object tracking called TPSR, which exploits some hand-crafted features, such as color, boundary, motion vector, etc. However, with the new development of deep neural network, many computer vision tasks have been adopted into this new trend. Convolutional neural networks (CNNs) have recently been applied to image classification [56], semantic segmentation [41], object detection [35] and etc. Such great success of CNNs is mostly attributed to their outstanding performance in representing visual data. Therefore, researchers start to investigate the possibility of applying CNN architecture for visual tracking problem.

One direction is that by extracting deep CNN features from pretrained neural network, such as AlexNet [56] and VGGNet [92], researchers are able to represent the target in a more discriminative way [42,71,109]. Following tracking-by-detection scheme, these features are imported into some traditional pipelines, such as correlation filters and SVM machines, to distinguish the target and background. This direction does not fully exploit the power of CNN and there is no end-to-end training since it is difficult to collect a large amount of training data for video processing. Although these methods may be sufficient to obtain generic feature representations, its effectiveness in terms of tracking is limited due to the fundamental inconsistency between tracking (locating targets of arbitrary classes) and classification (predicting object class labels) problems.

To fully exploit the advantages of CNN architectures, it is desirable to train them on large-scale data specialized for visual tracking, which cover a wide range of variations in the combination of target and background. However, it is truly challenging to learn a unified representation based on the video sequences that have completely different characteristics. Due to such variations and inconsistencies across sequences, researchers believe that the ordinary learning methods based on the standard classification task are not appropriate for visual tracking. Motivated by this fact, the multi-domain network (MDNet) tracker [84] is proposed to learn the shared representation of general targets from multiple annotated video sequences for tracking, where each video is regarded as a separate domain, and then online track and learn the representation of new target during the testing process. It consists of the shared layers, which are targeted for extracting generic object representation that invariants to environment factors, such as illumination and scale change, and domain specific layers for target localization. The MDNet trains end-to-end in offline learning process and tests in an online tracking with update learning fashion. Based on the MDNet tracker, [27] introduces a self-structure RNN representation into the network to distinguish the target out of the distractors during the tracking. [83] builds a branch of multiple CNN trackers for the target in a tree structure to manage multiple target appearance models.

Most of the state-of-the-art tracking system focus on building a spatial detector to find the target. For example, the MDNet tracker trains the convolutional neural network to classify the target out of the background region. However, it has several major weaknesses on handling situations like articulated motion, fast motion and distractors. In

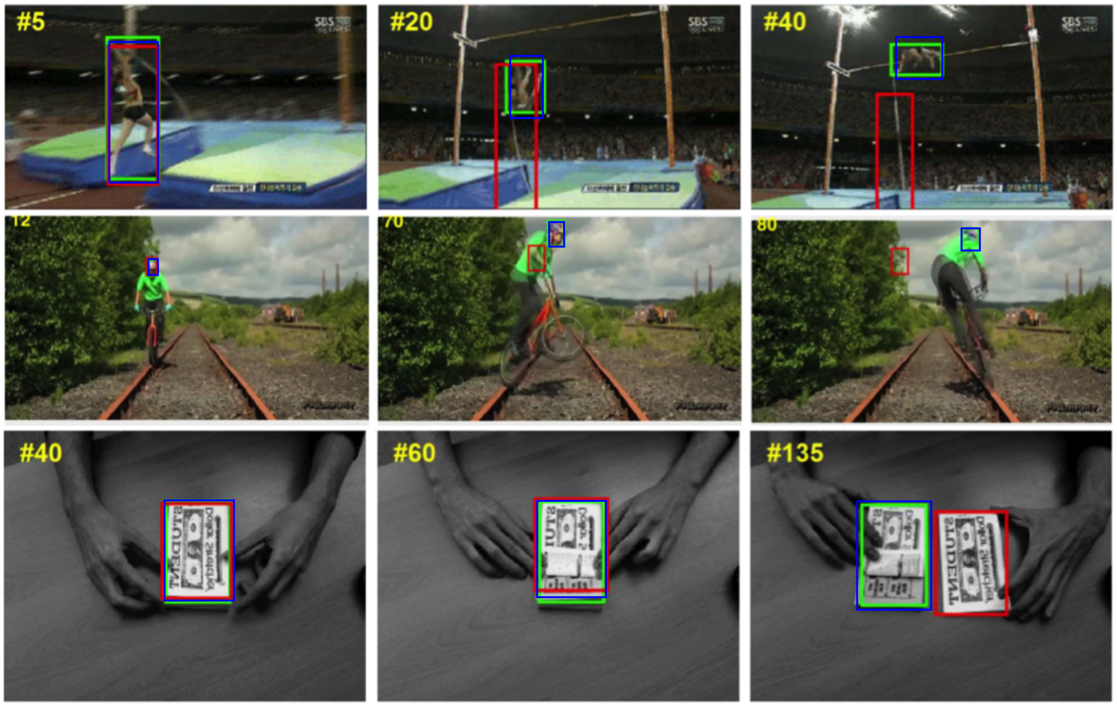


Figure 4.1: Performance comparison for the failure cases of the MDNet (Jump, Biker and Coupon). Green, red and blue bounding boxes denote the ground-truths, MDNet and MGNet tracking results, respectively.

Figure 4.1, it shows the comparison between our proposed motion-guided CNN tracker and the MDNet tracker. The reason why the MDNet fails is that the system cannot generate the correct candidate regions and the network is poor at handling different motion scenarios. To summarize the failure cases, they are:

- **Distractors.** Since the MDNet focuses on spatial representation of the target, it is very confusing when two similar objects are presented. The distractor is a trouble maker in the pipeline of the MDNet tracker.

- **Articulated motion.** Human movement is a common target in videos. Deformable parts, such as arm, leg and body, are difficult to track as one unit target. Therefore, this kind of articulated motion is very challenging for the MDNet.

- **Fast motion.** With fast motion, target in next frame is very far away from previous location. However, the search window of the MDNet is not capable of capturing the correct target in this scenario.

Therefore, in this work, an online object tracking system called motion guided convolutional neural network (MGNet) is proposed to enhance the motion handling ability in the original MDNet so that the system can successfully track the target for failure cases as listed above.

Overall speaking, there are two major contributions in this work. First, the use of dynamic motion model to generate the correct candidate regions is essential for tracking since if the candidates are incorrect, it is impossible to locate the target successfully. On another hand, an accurate target location estimation also reduces the number of candidates and speeds up the tracking process. Second, it is the first time that the spatial RGB and temporal optical flow are combined together as the network inputs to show the discriminative power of the tracking system. Optical flow map indicates the motion vector for each pixel between two adjacent frames, which provides important movement and segmentation cues for target localization. These two ideas enhance the motion handling ability of the original MDNet tracker.

The rest of this paper is organized as follows. Section 4.2 is a brief review of the multi-domain network (MDNet) tracker. Then, a detailed explanation of the proposed motion guided convolutional neural network (MGNet) tracker is presented in Section 4.3. Its overall tracking performance is shown and compared with the state-of-the-art methods against two widely used benchmarking datasets in Section 4.4. Section 4.5 provides the detail analysis about the contributions of each component in the proposed MGNet by comparing the tracking performance with the MDNet tracker. Finally, concluding marks are given in Section 4.6.

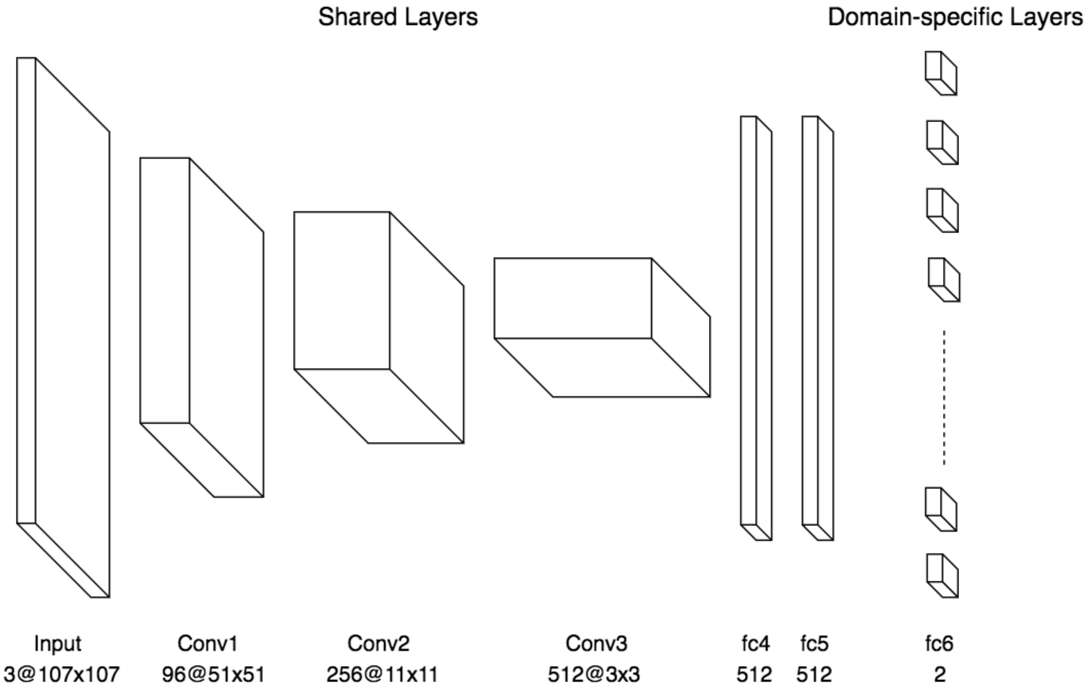


Figure 4.2: Network architecture of the multi domain network tracker.

4.2 Review on the Multi-Domain Network Tracker

The Multi-domain network tracker (MDNet) [84] is an effective tracking framework, which consists of multi-domain representation learning and online visual tracking. It learns the shared representation of targets from multiple annotated video sequences for tracking, where each video is regarded as a separate domain. The network has separate branches of domain-specific layers for binary classification at the end of the network, and shares the common information captured from all sequences in the preceding layers for generic representation learning. When a test sequence is given, all the existing branches of binary classification layers, which were used in the training phase, are removed and a new single branch is constructed to compute target scores in the test sequence. The new classification layer and the fully connected layers within the shared layers are then fine-tuned online during tracking to adapt to the new domain.

4.2.1 Multi-Domain Representation Learning

- **Network architecture.**

  The architecture of the MDNet is illustrated in Fig. 4.2. It receives a 107×107 RGB input, and has five hidden layers including three convolutional layers ($conv1 - 3$) and two fully connected layers ($fc4 - 5$). Additionally, the network has K branches for the last fully connected layers ($fc6^1 - fc6^K$) corresponding to K domains, in other words, training sequences. The convolutional layers are identical to the corresponding parts of VGG network [92] except for the feature map sizes. The next two fully connected layers have 512 output units and are combined with ReLUs and dropouts. Each of the K branches contains a binary classification layer with softmax crossentropy loss, which is responsible for distinguishing target and background in each domain.

- **Learning algorithm.**

  The goal of the learning algorithm is to train a multi-domain CNN disambiguating target and background in an arbitrary domain, which is not straightforward since the training data from different domains have different notions of target and background. However, there still exist some common properties that are desirable for target representations in all domains, such as robustness to illumination changes, motion blur, scale variations, etc. To extract useful features satisfying these common properties, MDNet separate domain-independent information from domain-specific one by incorporating a multi-domain learning framework. This CNN network is trained by the Stochastic Gradient Descent (SGD) method, where each domain is handled exclusively in each iteration.

For detail implementation of this learning procedure, each frame of all the sequences will generate 250 training samples (50 positive ones and 200 negative ones), where positive and negative samples have ≥ 0.7 and ≤ 0.5 IoU overlap ratios with ground-truth bounding boxes, respectively. Network is trained for 100K iterations with learning rates 0.0001 for convolutional layers and 0.001 for fully connected layers.

4.2.2 Online Visual Tracking

Once the multi-domain learning complete, the multiple branches of domain-specific layers ($fc6^1 - fc6^K$) are replaced with a single branch ($fc6$) for a new test sequence. Then the system fine-tune the new domain-specific layer and the fully connected layers in the shared network online at the same time.

- **Tracking control and network update.** To estimate the target state in each frame, N target candidates $\mathbf{x}^1, \ldots, \mathbf{x}^N$ sampled around the previous target state are evaluated using the network. The optimal target state $\mathbf{x}^*$ is given by finding the example with the maximum positive score as

$$\mathbf{x}^* = \arg\max_{\mathbf{x}^i} f^+(\mathbf{x}^i)$$

- **Hard minibatch mining.**

  The majority of negative samples are typically trivial or redundant in tracking-by-detection approaches, while only a few distracting negative samples are effective to training a classifier. Hence, the ordinary SGD method, where the training samples evenly contribute to learning, easily suffers from a drift problem. A popular solution in object detection for this issue is hard negative mining [99], where training and testing procedures are alternated to identify the hard negative samples. The MDNet adopts this idea in the online tracking procedure. As the learning proceeds and the network becomes more discriminative, the classification in a minibatch becomes more challenging as illustrated in Figure 4.3. This approach

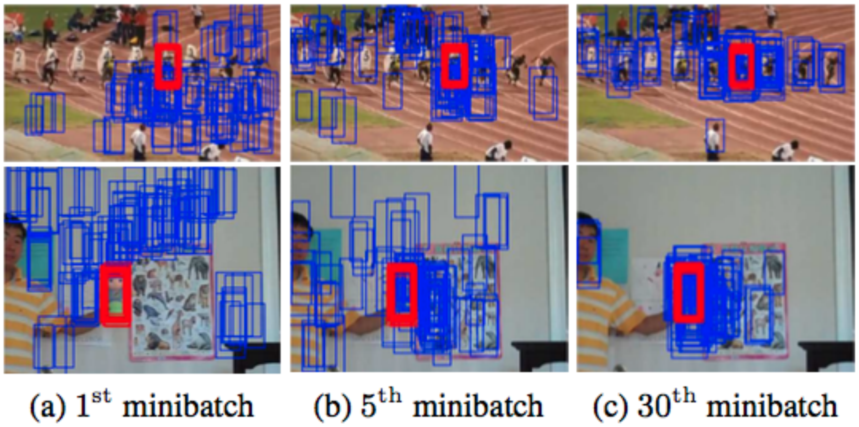


Figure 4.3: Identified training examples through hard negative mining in Bolt2 (top) and Doll (bottom) sequences. Red and blue bounding boxes denote positive and negative samples in each minibatch, respectively. The negative samples becomes hard to classify as training proceeds.

examines a predefined number of samples and identifies critical negative examples effectively without explicitly running a detector to extract false positives as in the standard hard negative mining techniques.

- **Bounding box regression.**

  Due to the high-level abstraction of CNN-based features and the data augmentation strategy which samples multiple positive examples around the target, the network sometimes fails to find tight bounding boxes enclosing the target. In order to improve the target localization accuracy, the bounding box regression technique [35] is applied and this linear regression model is trained using conv3 features of the samples near the target.

4.2.3 Summary

Experimental benchmark results show that the tracking performance of MDNet is the best among the current visual object trackers. It is the winner of 2015 Visual Object

Tracking challenging. It is very powerful in handling video sequences that contain illumination variation, scale variation and etc. However, during the study of this model, we still find out some drawbacks and that's why we are trying to solve them. More details are explained and discussed in the next Section 4.3.

4.3 The Proposed MGNet Tracker

The proposed MGNet is built upon the MDNet with two innovations: 1) motion-based candidate selection (MCS) using a dynamic prediction model, and 2) CNN with RGB-plus-motion (RGB-M) 5-channel input. The details are elaborated in the following two subsections.

4.3.1 Motion-based Candidate Selection (MCS)

The bounding box of a local region to be tested whether it is the desired location of an object is called a candidate. Candidate selection plays a critical role in the MDNet. It behaves like the object proposal in the object detection problem. If the system fails to generate appropriate candidates, it will not be able to find the target location correctly. However, the existing candidate selection strategy in the MDNet is very problematic for articulated motion where the aspect ratio of width/height is changing and fast motion where the search range is not big enough. In this scenario, system fails to generate correct candidate, which results that the tracker either loses or inaccurately tracks the target.

The proposed candidate selection is motivated by the classic model-based trajectory prediction framework such as those used in the Kalman filter [3] and the Markov Chain Monte Carlo prediction method [50]. These models often consist of a system state vector, motion dynamics modeling and description of observations (or measurements). However, these models cannot be applied directly to the current problem. Some modifications are required.

We are often interested in the first- and second-order statistics of the system state vector, which are the mean and the co-variance matrix of the random vector. If the state vector is multivariate Gaussian, the mean and the co-variance matrix can be used to determine the probability distribution of the state vector. In the proposed system, four parameters in the online object tracking problem are considered: the center location (x, y), width w and height h of the bounding box of candidates. Those parameters (x, y, w, h) form the state vector. To allow the maximum flexibility such as tracking of objects of articulated motion and observing from an arbitrary viewpoint, these four parameters are treated individually and independently. Without loss of generality, we use the horizontal location x of the center of a bounding box as an example in the following discussion. The same methodology applies to the other three parameters y, w and h.

For a given bounding box of a candidate characterized by (x, y, w, h), we determine its likelihood of being the target region based on the the output score of a trained MGNet or MDNet. If the score is higher, the candidate is more likely to be the desired target. Thus we choose the bounding box with the highest score as the final candidate. For this reason, we say that the network plays the role of observation evaluation.

Let x_t denote the state of the horizontal location of the target in frame t. Based on the Markov assumption, the true state is conditionally independent of all earlier states given the previous state.

$$p(x_t|x_1, \ldots, x_{t-1}) = p(x_t|x_{t-1}) \tag{4.1}$$

Similarly, the measurement z_t in frame t is dependent only upon the current state and is conditionally independent of all previous states given the current state.

$$p(z_t|x_1, \ldots, x_t) = p(z_t|x_t) \tag{4.2}$$

Therefore, the probabilility distribution of current state x_t given the previous observations $\mathbf{Z}_{t-1} = \{z_1, \ldots, z_{t-1}\}$ can be expressed as

$$p(x_t|\mathbf{Z}_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|\mathbf{Z}_{t-1})dx_{t-1}. \tag{4.3}$$

In order to solve this equation, two basic models are considered: observation measurement model and state transaction motion model. As discussed, the trained network plays the role of observation measurement. Here we would like to examine three different motion transaction models which are listed below. We use z_t^* to denote the optimal observation in frame t, which is the tracking location result.

1. **Zero velocity** (ZV) In this motion model, the distribution of the candidates in current frame is centered at the location in previous frame z_{t-1}^*, which means:

   $$\mathbb{E}[x_t|\mathbf{Z}_{t-1}] = z_{t-1}^* \tag{4.4}$$

   Notice that this is the model which is applied in the original MDNet tracker.

2. **Constant velocity** (CV) Different from the zero velocity model, the constant velocity one considers the speed of the target in previous state to predict the location in current state. Based on previous optimal observations, the velocity in previous frame $v_{t-1} = z_{t-1}^* - z_{t-2}^*$. Therefore, the expectation of the current state is

   $$\mathbb{E}[x_t|\mathbf{Z}_{t-1}] = z_{t-1}^* + v_{t-1} \tag{4.5}$$

3. **Constant acceleration** (CA) Besides of the velocity, the acceleration in the previous frame is exploited in this model: $a_{t-1} = v_{t-1} - v_{t-2}$. The expectation of the current state can be computed by

$$\mathbb{E}[x_t|\mathbf{Z}_{t-1}] = z^*_{t-1} + v_{t-1} + a_{t-1} \tag{4.6}$$

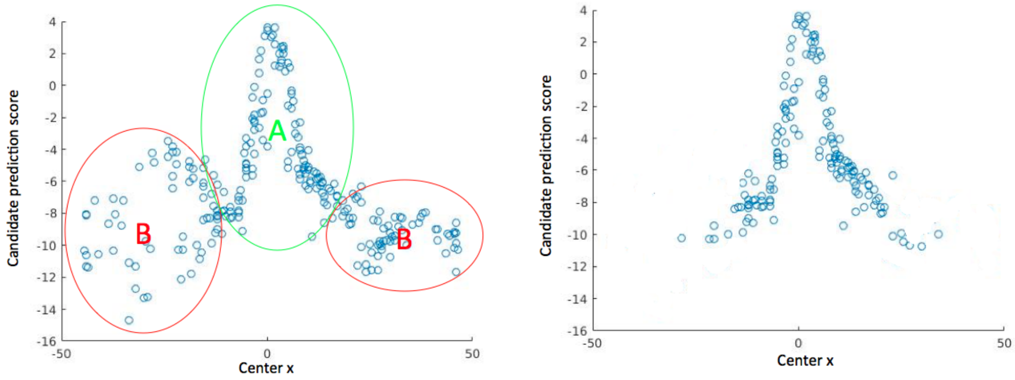
To validate the advantage of considering different motion types on expectation evaluation, Table 4.1 shows the mean of motion prediction error for all the sequences in the benchmark dataset [113]. Based on the ground truth of the target locations in previous frames, we calculate the predicted mean location of the target according to three different motion types. Then the motion prediction error between this mean and the ground truth locations for current frame is measured. Finally the averaged error is calculated from all the sequences. As we can see, the constant velocity model is more accurate to predict the target location. Therefore, in the proposed MGNet tracking system, the constant velocity model is adopted to predict the distribution of the current state.

Table 4.1: Overall prediction error for x (mean of all sequences)

	ZV	CV	CA
Mean prediction error (pixel)	3.2531	2.1766	3.7969

Besides of the expectation, variance is also a very important parameter to guarantee the accuracy and efficiency of the candidate selection as it indicates the degree of movement fluctuation of the target. The two independent variance estimations for width and height provide more convincing information to generate the correct candidate region.

The variance prediction is based on the observation score distribution of previous frame. Suppose we have N (=256) observations ($z^i_{t-1}, i = 1 \ldots N$) evaluated by the network in previous frame and each of them has been assigned with one score $s^i_{t-1}, i = 1 \ldots N$. If the candidate is close to the tracking location z^*_{t-1}, then it is likely to have a



(a) Whole candidate score distribution

(b) Valid candidate score distribution

Figure 4.4: (a) Scores for all the candidates (not good for Gaussian fitting). Set A is valid; Set B is not. (b) Valid candidate set after denoising, which is suitable for Gaussian curve fitting.

high score. Here we assume the Gaussian distribution and the curve fitting is applied to find the variance of the score distribution. Subsequently, the variance is used to generate the candidate set in current frame.

$$Var[x_t|\mathbf{Z}_{t-1}] = Var[z_{t-1}] \tag{4.7}$$

However, single Gaussian is not capable of providing accurate curve fitting for all the candidate scores. The main reason is that some candidates selected are far away from the target region, which usually belongs to background. In other words, the scores of those candidates become noise in curve fitting. Figure 4.4(a) shows an example of the network decision scores of all the candidates. It is clear to see that single Gaussian does not fit all the scattered scores. Therefore, a denoising procedure is necessary to separate the valid and invalid candidates for variance modeling.

Candidates in region A are labeled as valid ones since they are overlapped with the target region. Candidates in region B are considered as invalid ones since they are more likely related to background. The denoised result is shown in Figure 4.4(b).

4.3.2 RGB-plus-Motion 5-Channel Input (RGB-M)

In this part, we would like to enhance the discriminative power of the network decision ability by exploiting the motion optical flow cue. In the original MDNet, it uses RGB information to build the spatial target detector. However, the performance is not satisfactory in some cases. For a deformable object such as human, it is difficult to find a good spatial representation. In another case, when target and background clutter in the bounding box, the system cannot differentiate them successfully. Considering of these, we observe that the optical flow map between two adjacent frames provides valuable pixel-wised motion and segmentation cues for target localization, even for the cases with background clutter. Therefore, two optical flow channels (horizontal and vertical) are added as the input signal in the proposed MGNet tracking system. In other words, we integrate both of the spatial (RGB) and motion (optical flow) cues into the network and have great advantages from the combination. Figure 4.5 shows the network architecture of the proposed MGNet. The input signal is modified from three channels (3@107x107) to five channels (5@107x107). Correspondingly, the size of the filters in the first convolutional layers needs to be changed to 7x7x5.

Also, another network modification compared with the original MDNet is that the number of filters in the first convolutional layer is enlarged from 96 to 112. As we can see in the first CONV layer, it combines both the spatial and motion cues through convolution operation. Because of adding the optical flow signal, we believe that the network needs a larger filter bank (higher feature space) to effectively present its discriminative power. This idea is motivated by the ”REctified-COrrelations on a Sphere” model in [57, 58]. It shows that after the CNN training process, the converged filter weights define a set of anchor vectors in the RECOS model where the anchor vectors represent the frequently occurring patterns of the training data. In the original MDNet,

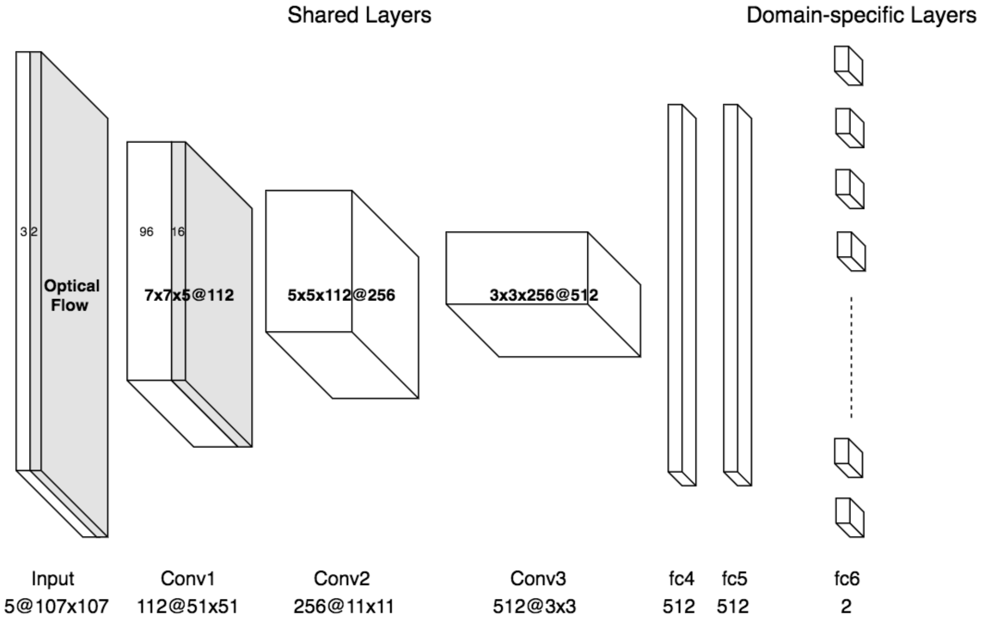


Figure 4.5: Network architecture of the proposed MGNet.

it simply represents the RGB patterns. With additional optical flow channels, the patterns now are spatial-motion combined, which means the feature space becomes larger. Therefore additional filters are required to describe the spatial-motion patterns. The final number of filters is determined by a standard algorithm called tree structured K-means clustering.

In order to prove the correctness of the above idea, a verification experiment is conducted. In the experiments, the only modification in the original MDNet is the number of filters in the first convolutional layer, which means the candidate selection keeps the same. We report the number of filters versus the final converged training classification error in Figure 4.6. This error measures the averaged detection and tracking errors after the network training. As can be seen from the figure, the convergence error is 0.041 with original filter number 96 in the first layer. When the number of filters reaches around 112, the error is converged to 0.03.

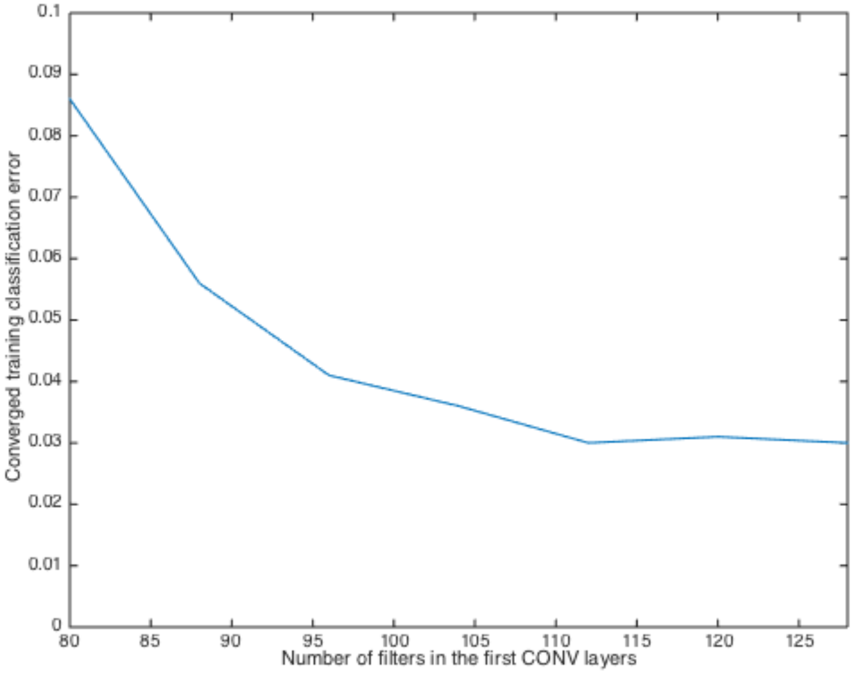
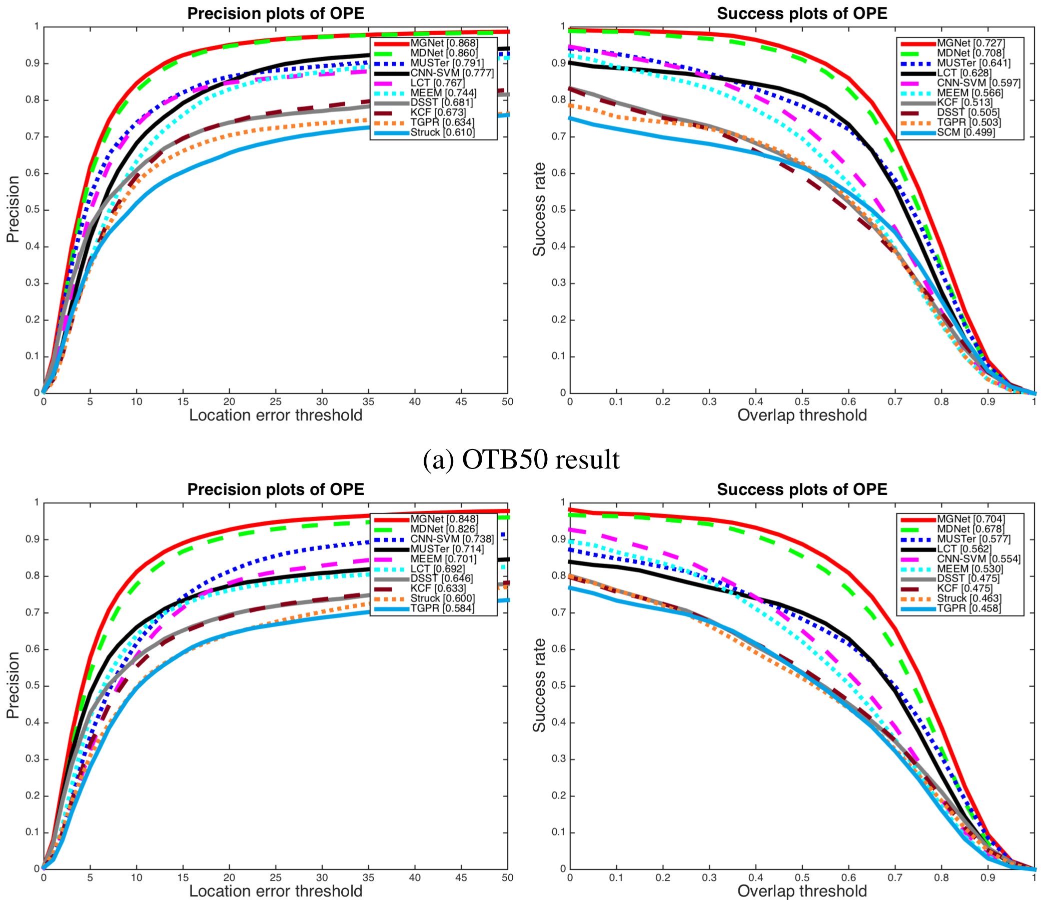


Figure 4.6: Number of filters in first CONV layer V.S. converged training classification error.

4.4 Experimental Results

The performance of the proposed motion guided convolutional neural network tracker is evaluated with quite a few recent state-of-the-art algorithms against two well known datasets: Object Tracking Benchmark (OTB) [112, 113] and Visual Object Tracking challenge dataset (VOT2015) [54]. The implementation is based on original MDNet [85] using MatConvNet toolbox [106]. All the optical flow maps [98] are calculated and stored in memory before training and testing. The experiments and evaluations are conducted with Intel Core i7-5930K CPU @ 3.50GHz ×12 and GeForce GTX TITAN X GPU.



(a) OTB50 result

(b) OTB100 result

Figure 4.7: The location precision and success rate performance curves of top 10 tracking algorithms on OTB50 and OTB100. The AUC value of the MGNet method ranks No. 1 both in the location precision plot and success rate plot.

4.4.1 Object Tracking Benchmark (OTB) Dataset

There are 100 video sequences in OTB with various challenging attributes such as deformation, rotation, fast motion, etc. In this experiment, the network is trained by Visual Object Tracking challenge dataset (VOT2015) [54], which contains 61 video sequences.

Overall performance. According to the evaluation methodology in [112], we use both of these two metrics: the center location error and the overlap success rate to compare the tracking performance. The former measures the averaged Euclidean distance between the centers of the ground truth and the tracking bounding box. The latter is the percentages of frames where the overlapping region between the ground truth and the tracking bounding box surpasses a threshold.

Figure 4.7 shows the overall tracking performance of the proposed motion guided convolutional neural network (MGNet) tracker on OTB50 [112] and OTB100 [113]. We compare the performance of more than 30 methods and show the results of the top ten. They are the proposed MGNet tracker, MDNet [85], CNN-SVM [42], MUSTer [43], MEEM [124], LCT [72], DSST [22], KCF [39], Struck [38] and TGPR [33]. The area under the curve (AUC) values of the proposed MGNet tracker rank No. 1 in terms of location precision and success rate for two datasets (OTB50, OTB100). If the proposed method is compared with MDNet, the performance improvement on OTB50 dataset are 0.8% and 1.9% for precision and success, respectively. The OTB100, where the additional 50 video sequences contain attributes such as fast motion, articulated motion, and distractors, is more challenging to MDNet. The performance of the proposed MGNet shows larger improvement with 2.2% in precision and 2.6% in success on this dataset. This is benefited from considering motion models in the proposed system.

Attribute-based Evaluation. In the benchmark dataset, all video sequences are labelled with eleven attributes. Figure 4.8 demonstrates the success rate for nine main attributes. The other two are omitted because the number of sequences belonging to these two categories is small and the proposed MGNet also outperforms other methods. As shown in those figures, the MGNet has better tracking results than all the others. Particularly for attributes like deformation and background clutter, the performance improvement is obvious: 4% and 3.7%, respectively from the original MDNet. On one hand, the candidate selection scheme from dynamic motion model provides a more accurate prediction of the movement of the target, including the direction and self-rotation. Therefore, compared with the original MDNet tracker, MGNet is more

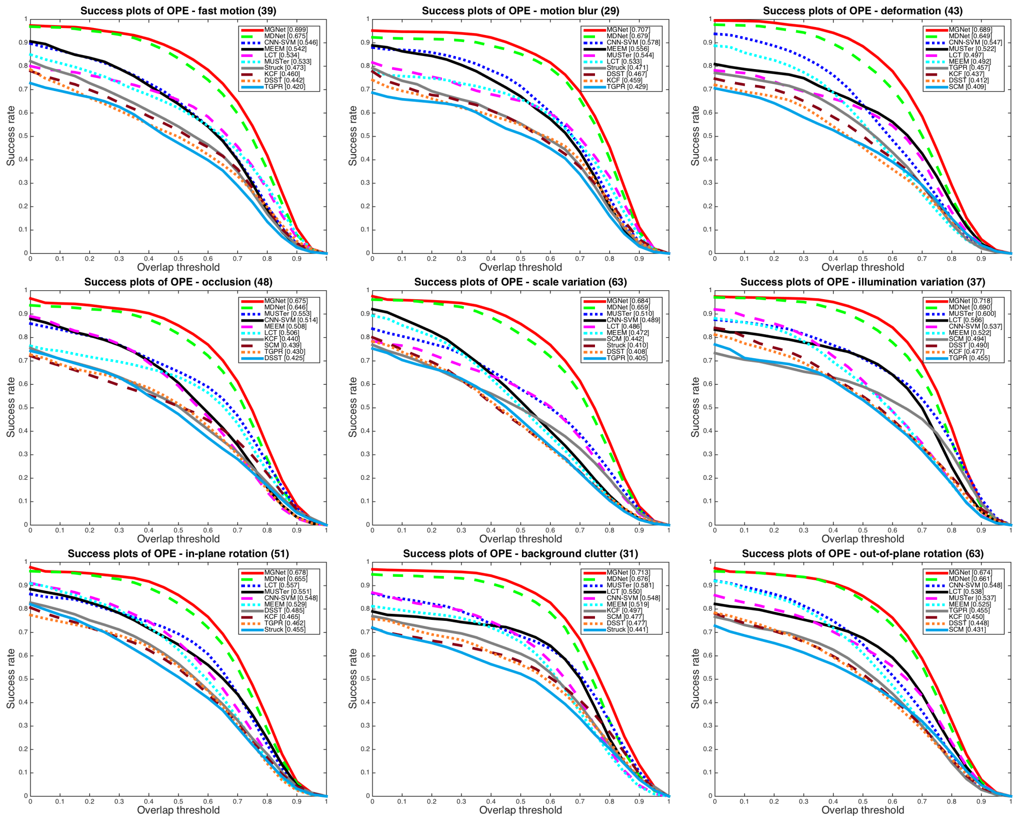


Figure 4.8: Success rate plots with respect to night attributes of video sequences: fast motion, motion blur, deformation, occlusion, scale variation, illumination variation, in-plane rotation, background clutter and out-of-plane rotation.

powerful to track the deformable targets. On another hand, because of the enhanced discriminative power of the network from pixel-wised motion cue, the network is able to differentiate the target in the bounding box even though the background clutters. From this attribute-based evaluation, we can obviously see the strength of the proposed motion guided CNN tracker.

Qualitative evaluation. Eight representative sequences are chosen from the benchmark dataset in [113] to show the tracking results of six algorithms in Figure 4.9 for qualitative evaluation. They are the proposed MGNet, MDNet, CNN-SVM, LCT, MUSTer and MEEM.

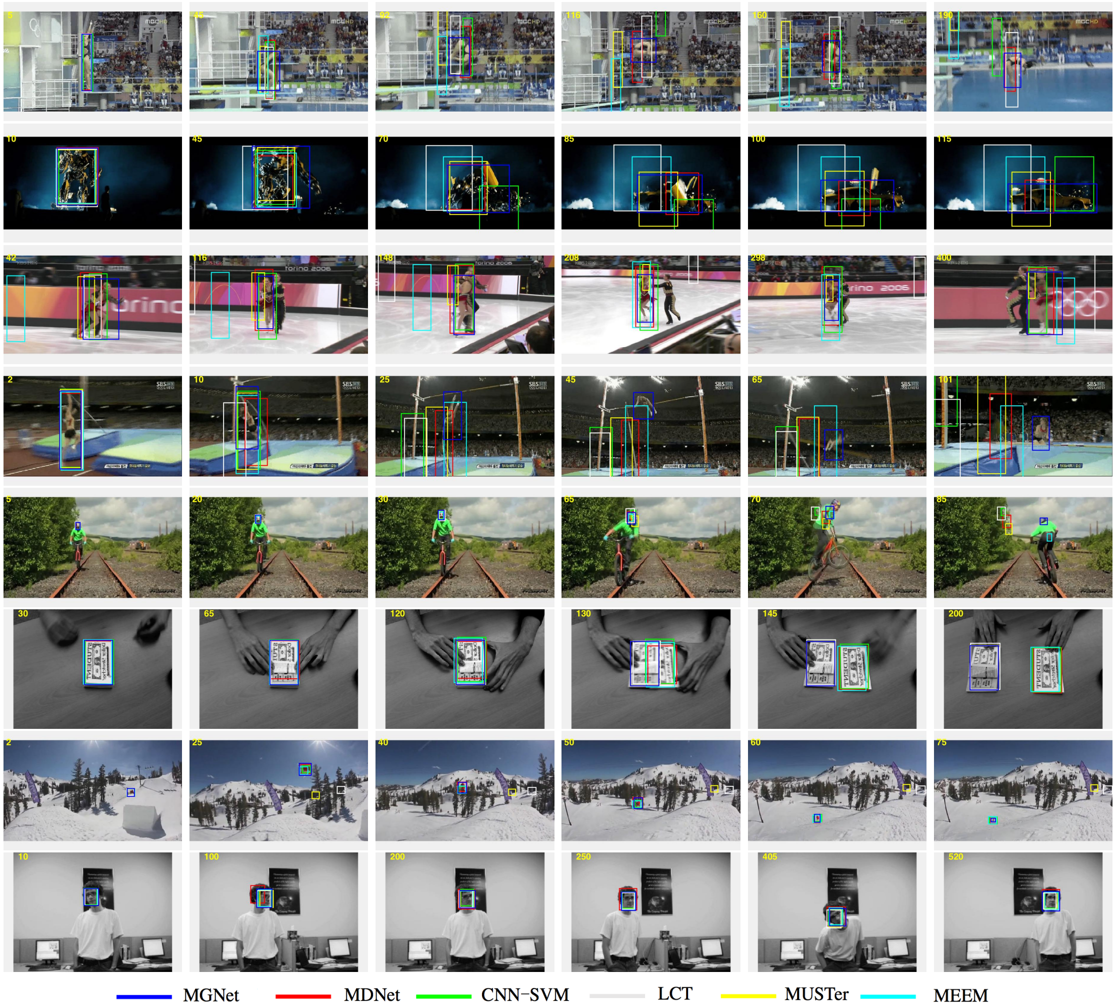


Figure 4.9: Performance visualization of proposed MGNet, MDNet, CNN-SVM, LCT, MUSTer and MEEM methods on eight representive sequences (top to down are Diving, Trans, Skating2-1, Jump, Biker, Coupon, Skiing and David2, respectively).

The first three sequences in Figure 4.9 are Diving, Trans and Skating2-1. Targets in them are all deformable objects. From the visual results, we can clearly see that the proposed MGNet tracker outperforms others in accurately predicting the aspect ratio of the bounding box of the target, which means the success overlap score is much higher. The fourth sequence is Jump, which is combined with fast and articulate motions. For all the other trackers, they lost the target in the middle of the sequence. However, the MGNet tracker follows the target successfully till the end of the video clip. This shows the strength of our motion prediction. The fifth sequence is Biker with large motion. We can see even though the proposed MGNet tracker seems to miss the target in frame 70, it is still the closest one to the ground truth location compared with others. More importantly, the failure detection scheme in the proposed system can send an alert so that the target can be re-captured in the future frames. The sixth one is Coupon, which shows the advantage of MGNet tracker in handling distractor. Because of the guide from motion movement, our tracker is able to correctly follow the target. The last two clips are Skiing and David2, which demonstrate that the MGNet tracker outperforms the MDNet tracker to find the target location with higher accuracy.

4.4.2 Visual Object Tracking (VOT) Challenge Dataset

Visual object tracking challenge (VOT2015) [54] is another commonly used dataset for benchmarking, which contains 61 sequences. Similar to OTB100dataset, it focuses on generic object target. In this section, the MGNet tracker is trained with OTB100 and tested on VOT2015 dataset.

One new feature in VOT2015 dataset is the ground truth labeling. Different from the OTB100 where the target location is marked with a regular bounding box, VOT2015 allows the bounding box to be rotated. Also, besides of the accuracy measurement, VOT2015 adopts another robustness score to evaluate the performance of different trackers. The robustness score is defined as the number of times the tracker fails in tracking in one individual sequence. Once the tracker drifts off the target, the system detects a tracking failure and will re-initializes the tracker. This re-initialization is triggered when the overlap score (intersection-over-union) drops to zero. There are two types of experiment settings: initialized/re-initialized with either ground truth bounding boxes (baseline) or randomly perturbed ones (region noise). For more details, please refer to [54].

Here we compare the proposed MGNet tracker with six state-of-the-art methods. They are MDNet [85], MUSTer [43], MEEM [124], DSST [22], KCF [39] and Struck [38]. Table 4.2 and 4.3 show the average scores and ranks of accuracy and robustness in VOT2015 [54]. The proposed MGNet method ranks the top both in accuracy and robustness evaluations.

Table 4.2: VOT2015 baseline evaluation

Tracker	Accuracy		Robustness		Expected overlap
	Score	Rank	Score	Rank	
MUSTer	0.52	4.28	2.00	3.77	0.19
MEEM	0.50	6.79	1.85	5.61	0.22
KCF	0.48	3.16	1.95	3.89	0.19
DSST	0.54	4.77	2.56	2.96	0.17
Struck	0.47	5.52	1.61	4.12	0.25
MDNet	0.60	2.89	0.69	2.06	0.38
MGNet	0.61	2.11	0.63	1.79	0.39

Table 4.3: VOT2015 region noise evaluation

Tracker	Accuracy		Robustness		Expected overlap
	Score	Rank	Score	Rank	
MUSTer	0.50	5.67	2.80	4.23	0.18
MEEM	0.48	6.32	2.19	5.31	0.20
KCF	0.49	4.16	2.35	3.12	0.18
DSST	0.52	3.98	3.56	3.22	0.17
Struck	0.49	5.88	2.61	4.98	0.27
MDNet	0.57	3.11	0.98	2.86	0.35
MGNet	0.58	2.89	0.88	2.98	0.36

4.5 Further Analysis and Discussion

As discussed, the proposed motion guided convolutional neural network (MGNet) tracker is based on multi-domain network (MDNet) tracker with two major modified components: dynamic motion model for candidate selection and additional optical flow motion signal for input. To show the contribution of each component, we would like to present the analysis by comparing MGNet with MDNet on the commonly used object tracking benchmark (OTB100) dataset [113] and VOT2015 dataset [54].

4.5.1 Impact of Added Components

One of the major modifications in MGNet tracker is the dynamic motion model for candidate selection. Rather than using the zero velocity model in the original MDNet tracker, we apply the constant velocity model with variance prediction into the network. Therefore, we would like to verify two advantages of this new candidate selection scheme: (1) find more accurate candidates (center location and aspect ratio of the bounding box); (2) reduce the number of selected candidates to speed up the tracking process. The experiment setup here is candidate selection module only, without optical flow input.

Table 4.4 shows the comparison of accuracy/speed versus the number of selected candidates. The accuracy here is from success overlap score plot with threshold T=0.8. The reason not to use center location error is that success overlap accuracy is more appropriate to indicate accurate tracking. Also, T=0.8 is a very high standard where the original MDNet tracker reaches only 0.38 in Figure 4.7. In Table 4.4, 256 is the default number of candidates in MDNet. If the same number of candidate is selected in MGNet, the accuracy is higher (0.43) while the processing time increases (1.42 s/f) due to the motion model. From another point of view, the performance of MGNet remains similar accuracy (0.38) by considering only 64 selected candidate, which means the computation speed (0.63 s/f) is much faster than MDNet (1.2 s/f). The strength of the proposed candidate selection scheme is clearly demonstrated. One thing to be mentioned here is that we use N=256 for performance comparison in this paper.

Table 4.4: Accuracy/Speed(second/frame) V.S. Number of Candidates: (*) refers to the original MDNet.

N	256(*)	256	128	64	32	16
Accuracy	0.38	0.43	0.40	0.38	0.25	0.12
Speed (s/f)	1.2	1.42	1.02	0.63	044	0.26

To verify the contribution of each component in the proposed MGNet tracker, we report the internal comparison of three different structures.

- Baseline: MDNet tracker
- MDNet + Candidate Selection (CS)
- MGNet: MDNet + Candidate Selection + Optical Flow

In Figure 4.10, we can see that the modified candidate selection scheme improves the original MDNet by about 1-2%. In this motion prediction model, it provides more information of the target region, such as the mean location and dynamic movement degree, for the system to select accurate candidates. Optical flow signal enhances the discriminative power of the network to more accurately determine the candidate positive score in the situations like background clutter and fast motion. The performance gain of the optical flow signal from the candidate selection is around 1%, as shown in Figure 4.10.

4.5.2 Evaluation Based on Sequence Attributes

The proposed motion guided convolutional neural network tracker is based on the original MDNet tracker. After checking the performance of different components in previous two sections, now we would like to present another sequence-level evaluation. In this section, all the sequences are analyzed in the benchmark OTB100/VOT2015 datasets

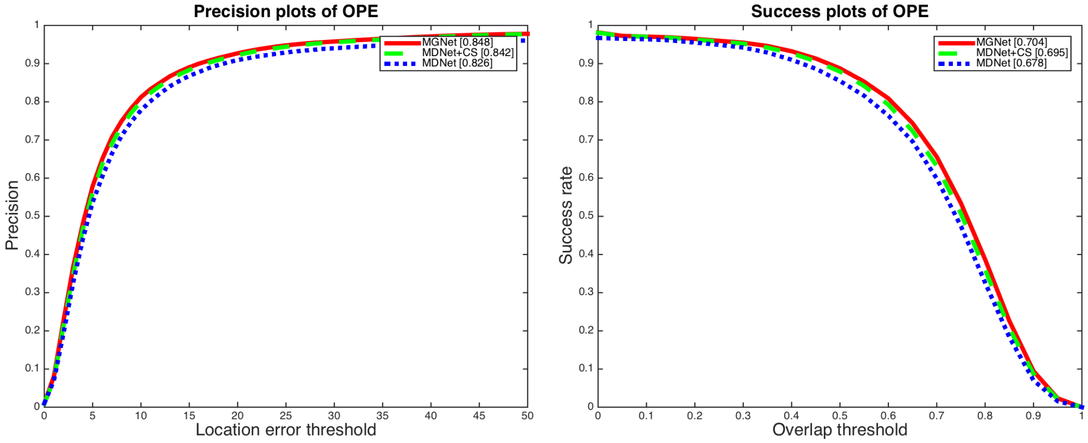


Figure 4.10: Precision and success plots for internal comparison on OTB100.

and the performances between MGNet tracker and MDNet tracker are compared with each other.

OTB100 Dataset

The sequences are grouped into three categories based on the tracking performance: (1) MGNet is better than MDNet; (2) MGNet is similar to MDNet; (3) MGNet is worse than MDNet. The grouping criterion is based on the overlap success plot with threshold T=0.7 since the success accuracy is higher than 0.8 if T=0.6 and it is lower than 0.5 if T=0.8. In other words, the performance change is obvious in the range of [0.6, 0.8]. After the threshold selection, we compare the success accuracy (SA) for each individual sequence between MGNet and MDNet trackers.

$$\begin{cases} \text{MGNet is better,} & \text{if SA(MGNet)-SA(MDNet)} > 0.06 \\ \text{MGNet is worse,} & \text{if SA(MGNet)-SA(MDNet)} < -0.06 \\ \text{MGNet is similar,} & \text{others} \end{cases}$$

MGNet is better than the MDNet. In this group, we target on the sequences where the proposed MGNet tracker works better while the original MDNet either fails (lose the target) or inaccurately tracks (wrong aspect ratio) the object. By analyzing this group, the strength of each modified component is clearly demonstrated. In Figure 4.11, it shows the precision and success plots for this group of sequences. The performance gain is large which is around 7-8%. By checking the properties of these sequences, it is interesting to find that each sequence in this group contains at least one of the following attributes.

- Articulate motion. Deformable object is one common type of targets in the video. One typical example is human (sequences like Diving, Jump). In the original MDNet, deformable parts, such as arms, legs and torso are difficult to track as one unit because of the complicated parts relation and the foreground/background clutter within the rectangle bounding box. MDNet is a spatial detector which is poor at tracking this kind of target since the spatial information becomes very confusing when there are changes of internal parts and background clutter. On the contrary, the optical flow motion signal in the proposed MGNet tracker helps the system to differentiate the target out of the background by exploiting the pixel-wised segmentation cue. Moreover, the MDNet assumes that the change variance of width and height is the same, which is not capable of dealing with cases with large target rotation. In the proposed system, we model the width and height separately so that the bounding box of the candidate can be more flexible with various aspect ratio in order to obtain a more accurate result.

- Fast motion. Objects with fast motion are also very challenging for the original MDNet tracker. With this attribute, target in the next frame is very far away from the previous location (sequences like Jump, Biker). The zero velocity model which uses the previous location as the searching center is adopted in MDNet.

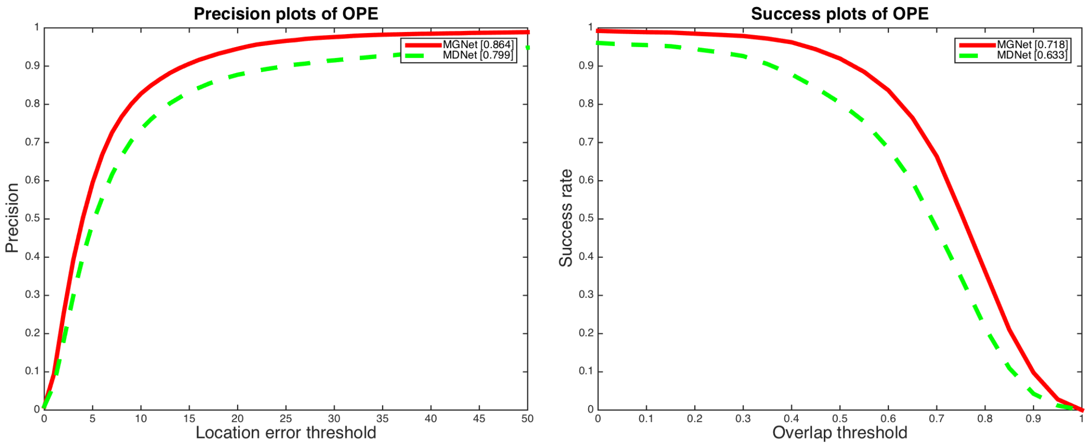


Figure 4.11: Precision and success plots for sequence group 1 in OTB100 (40 out of 100): MGNet is better.

That means it fails to find the correct target region because of the inappropriate search window. However, the constant velocity model proposed in MGNet is able to predict the next location by exploiting previous movement information so that the search window is more likely to overlap the target region.

- Distractor. Distractor means that some objects are similar to the real target (sequences like Coupon) so that the tracker is easy to get confused and the result is drifted to another object and never be resumed back. The MDNet tracker is a spatial pattern detector so that it is not able to deal with this case properly. However, with the help of motion cues, the MGNet can identify the correct target. Both the motion prediction model and optical flow cue assist the system to memorize the movement information of the target. In other words, the motion model guarantees the target is located in the bounding box and the optical flow cue enhances the discriminative power of the network to find the correct target.

In this group, there are 40 out of 100 sequences in total. Table 4.5 lists several representative sequences with three major attributes labeling.

Table 4.5: Sequences with attributes labeling in OTB100 (A: Articulated motion; F: Fast motion; D: Distractor).

Sequence	Attributes	Sequence	Attributes
Basketball	A, D	Biker	F
Blurbody	A	Bolt	A, D
Bolt2	A, D	Couple	A, F
Diving	A	Jump	A, F
Jumping	F	Motorrolling	A
Skating2	A	Walking2	D
Coupon	D	Gym	A, F
Skater	A	Skater2	A
Trans	A	Dancer	A
Dancer2	A	Liquor	D

MGNet is similar to the MDNet. As described, the tracking performance gap of each sequence in this group is smaller than 0.05 with overlap threshold T=0.7. There are 56 out of 100 sequences in this group. The original MDNet tracker is a spatial target detector based on RGB color information. In our proposed MGNet tracker, even though the input is a 5-channel signal from RGB and optical flow, the spatial cue is retained powerful to find the target region. The optical flow cue increases the discriminative power of the network when the spatial cue does not work well as discussed in previous section. However, the temporal motion information is not always helpful. One scenario is about the static target with only camera movement. In this case, the optical flow map indicates the global camera motion, which does not provide any object motion cue for segmentation. Then system still relies on spatial pattern detector. Another scenario is that both the movement of the target (deformable) and background are very complex that sometimes the optical flow map will be confused. So summarize for this group, we can see that for most of the cases, our proposed tracking system preserves the functionality of spatial detector without too much damage from the additional temporal motion cue.

MGNet is worse than the MDNet. In this group, 4 out of 100 sequences that the tracking performance of the MDNet tracker is better than the proposed MGNet tracker.

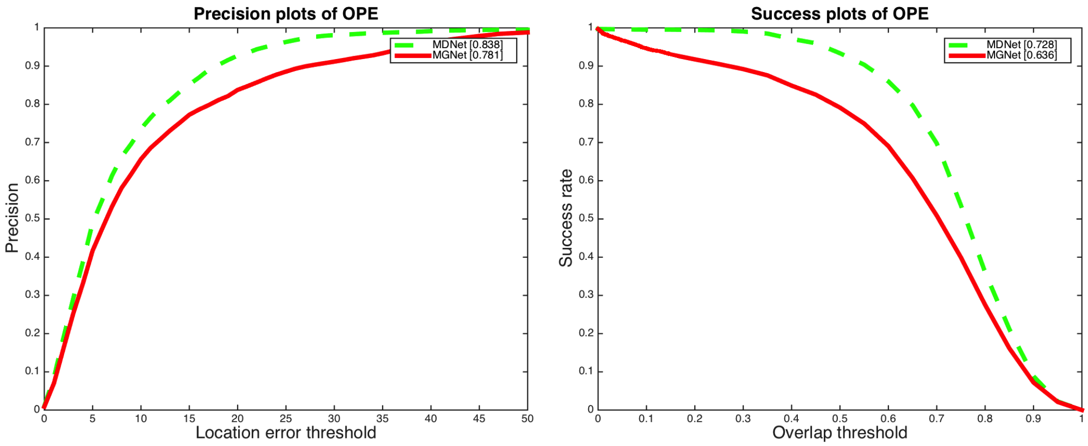


Figure 4.12: Precision and success plots for sequence group 3 in OTB100 (4 out of 100): MGNet is worse than the original MDNet.

These 4 video sequences are: Woman, FaceOcc2, Rubik, and Twinnings. The error analysis is detailed in the followings to have a better understanding of the weaknesses of the proposed MGNet tracker. First, we would like to report the center error precision and overlap success plots in Figure 4.12. For these sequences, the performance drop of the proposed MGNet tracker is 5.7% for center location error and 9.2% for overlap success rate. Since there are only 4 sequences in this group, which is a small proportion of the database, the proposed system still reach an obvious improvement on average. Moreover, Figure 4.13 shows these four sequences to realize this performance drop. Generally speaking, the reasons of the drop in these 4 sequences can be summarized into two categories:

- Occlusion. The first two sequences in Figure 4.13 are Woman and FaccOcc2. There are some partial occlusions in both of them and our proposed MGNet tracker does not perform well compared with the MDNet tracker. Take Woman sequence for example. Both of the MGNet and MDNet trackers works well at the beginning (frame 102). At frame 122, when there is a car partially occluding the target, the proposed MGNet tracker selects the visible region of the target only while the bounding box of the MDNet is closer to the labeled ground truth. When the occluded part shows again in frame 170, MGNet resumes to the correct region. Similar situation happens in frame 215 and 255. There are two explanations for this phenomenon: (1) The modified candidate selection scheme tries to find the accurate candidates. However, it is difficult to say whether the occluded part belongs to the target region or not since it depends on the problem/dataset definition. At least, the MGNet selects out the meaningful and reasonable candidates including the bounding boxes with and without the occluded part. (2) For the selected candidates, the network needs to determine the positive score for each bounding box. Since there is an additional optical flow map as the input, the decision making process in MGNet is more sensitive to the boundary segmentation from the motion cue. It means that the system is more likely to treat the occluded region (the vehicle) as the background region and selects a smaller bounding box as the target location. Similar analysis can be applied to the FaceOcc2 sequence, where the face is partially occluded by a book.

- Out-of-plane rotation with complex content. The other two sequences are Twinnings and Rubik. Both of them contain the out-of-plane rotation and the spatial content is complex. For sequence Twinnings, the locations of MGNet and MDNet in frame 180 are both correct. Starting from frame 190 (the second figure), the target tilts forward. In frame 195 (the third figure), the proposed MGNet tracks the original front view of the target rather than the whole target region. This is because the system believes that the front view region is closest to the target region in the initialization. From our points of view, this tracking result is reasonable and acceptable. Also, as can be seen from frames 200 and 220, when the front view region of the target is gone, the system claims that there is no correct region. In

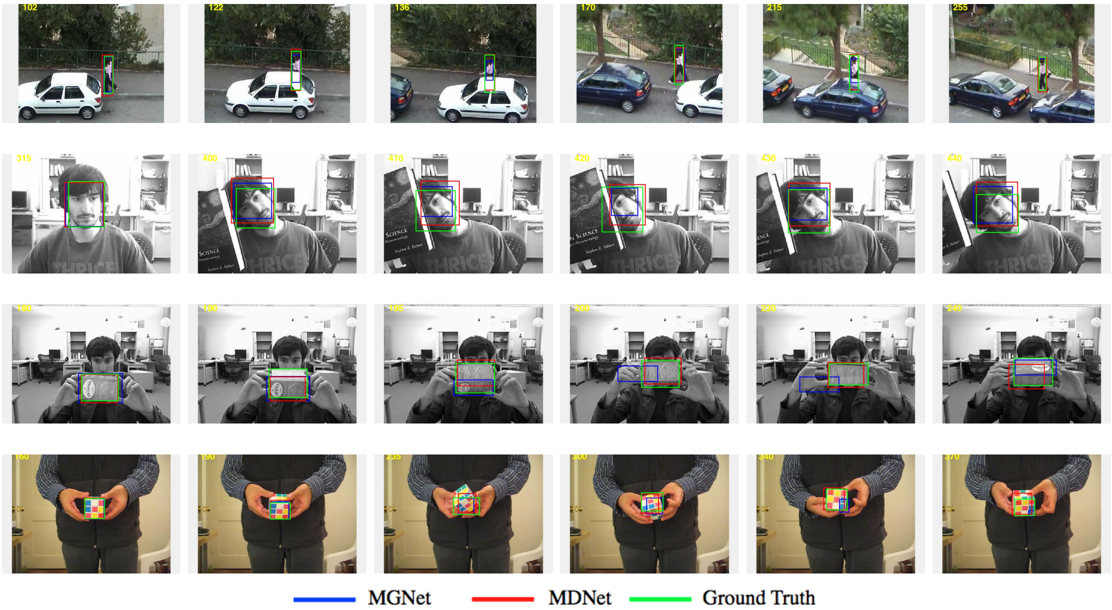


Figure 4.13: Performance visualization of proposed MGNet and MDNet methods on four representative sequences in group 3 of OTB100 (top to down are Woman, FaceOcc2, Twinnings and Rubik, respectively).

the last figure (frame 240), the target tilts back and the MGNet tracker re-detects the front view region when it appears.

VOT2015 Dataset

For completeness, the sequence-level comparison analysis for visual object tracking challenge (VOT2015) dataset is presented in this section. In this dataset, there are 61 video sequences, where a very small proportion of it are also in OTB100 dataset. The grouping strategy for this dataset is based on the accuracy score of each sequence. Table 4.6 demonstrates the analysis comparison.

In group ”MGNet is better”, similar improvement of the proposed MGNet can be found since it is better in handling articulated motion, fast motion and distractors. Figure 4.14 shows several representative sequences (Octopus, Bmx, Butterfly and Gymnastics1) in VOT2015. It is clear to see that for cases with object deformation, the proposed

Table 4.6: Sequence-level comparison with grouping for VOT2015 dataset

	MGNet is better	Two are similar	Challenge
# of sequence	17	41	3
Accuracy(MDNet)	0.61	0.60	0.42
Accuracy(MGNet)	0.63	0.60	0.45

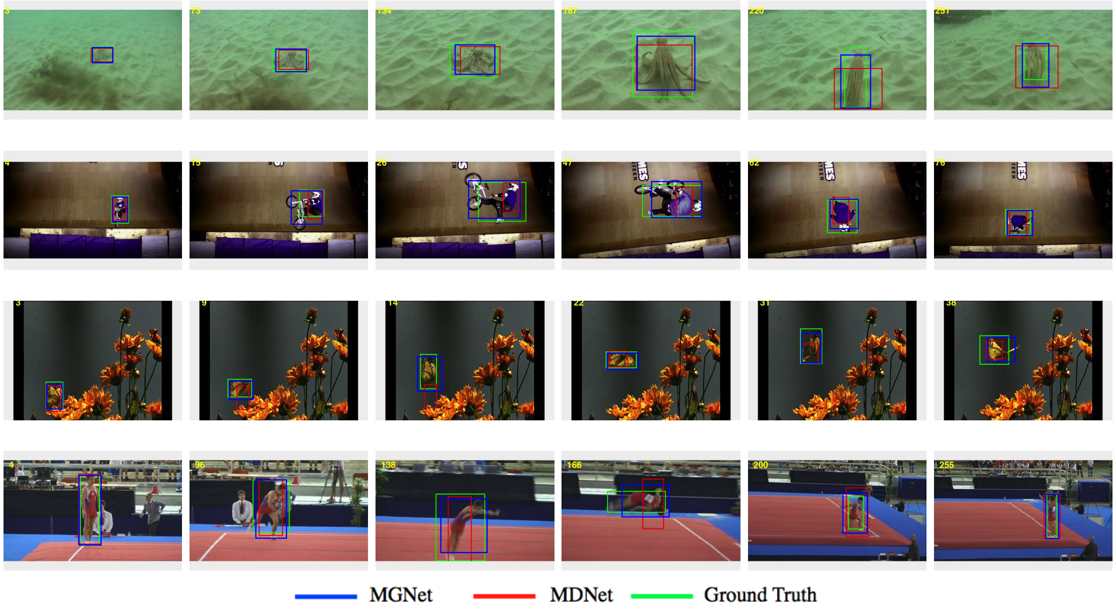


Figure 4.14: Performance visualization of proposed MGNet and MDNet methods on four representative sequences in group ”MGNet is better in VOT2015” (top to down are Octopus, Bmx, Butterfly and Gymnastics1, respectively).

MGNet tracker outperforms the MDNet tracker because of two modifications: dynamic model for candidate selection and additional optical flow input signal. One thing we should also notice is that for the third frame of Butterfly sequence, the MDNet tracker already misses the target while MGNet still works well. Later on, if the system detects that re-initialization is necessary, the MGNet can locate the target region accurately since it adopts the optical flow signal as the motion segmentation cue.

Most of the sequences still fall into the second group where the tracking performance of the proposed MGNet is similar to MDNet. There are 41 out of 61 sequences that the tracking performance of the proposed MGNet tracker is similar to the original MDNet.

For the third group, we name it as ”Challenging” sequences instead of ”MGNet is worse than the MDNet”. With the setup of the VOT dataset, if there happens severe occlusion where the tracker cannot access to the visible target region and the location drifts away from the ground truth, the system re-initializes. In other words, it is not able to differentiate the occlusion and tracking failure. From our experiment, even though the performance of MGNet is not as good as MDNet, both of them have low accuracy score and high robustness failure number. Therefore, we group several sequences such as Fish 2,3 and Leave as ”Challenging” sequences because they have high degree of deformation and complex foreground/background clutters where MGNet and MDNet cannot track well.

4.6 Conclusion

To conclude this chapter, in this work, a motion guided convolutional neural network (MGNet) tracker is proposed for online visual tracking based on the multi-domain network (MDNet) tracker. With two important modifications, the tracking results are improved significantly. These two modifications are the dynamic motion model for candidate selection and the spatial-motion cue as input to the system. For candidate selection, the proposals generated by the proposed system is more reasonable and accurate. Taking the optical flow map as input further enhances the discriminative power of the network. Through large scale data benchmarking evaluation, the advantages of the proposed method in handling sequences with articulated motion, fast motion and distractors are clearly demonstrated. However, there are still some limitations of the proposed MGNet method. One example is the partial occlusion issue which was explained before. The improvement of the boundary region between the target and the occluding object is helpful. Also, it is interesting to extend the single object tracker (SOT) to the multiple objects tracking (MOT) problem where the MOT problem is very challenging because of interaction and occlusion among multiple targets.

Chapter 5

Online CNN-based Multiple Object Tracking with Enhanced Model Updates and Identity Association

Online multiple objects tracking (MOT) is a challenging problem due to occlusions and interactions among targets. An online MOT method with enhanced model updates and identity association is presented to handle the error drift and the identity switch problems in this work. The proposed MOT system consists of multiple single CNN(Convolutional Neural Networks)-based object trackers, where the shared CONV layers are fixed and used to extract the appearance representation while target-specific FC layers are updated online to distinguish the target from background. Two model updates are developed to build an accurate tracker. When a target is visible and with smooth movement, we perform the incremental update based on its recent appearance. When a target experiences error drifting due to occlusion, we conduct the refresh update to clear all previous memory of the target. Moreover, we introduce an enhanced online ID assignment scheme based on multi-level features to confirm the trajectory of each target. Experimental results demonstrate that the proposed online MOT method outperforms other existing online methods against the MOT17 and MOT16 benchmark datasets and achieves the best performance in terms of ID association.

5.1 Introduction

The multiple objects tracking (MOT) technique predicts locations of multiple objects and maintains their identities to yield their individual motion trajectories throughout a video sequence. It has many applications such as video surveillance, human-computer interface and autonomous driving. However, it is a very challenging problem. This is especially true for sequences with frequent occlusions and interactions among targets in crowded scenes. The tracking-by-detection strategy is one of the most common ideas in various tracking tasks, where the impressive performance improvement comes from the development of a powerful object detector. For this reason, the MOT challenge [63, 78], which is the most popular MOT benchmark dataset and aims at multiple pedestrian tracking, provides all targets detection results in each frame directly. In other words, the initialization of target locations is not human-labelled but purely dependent upon detection results. Then, the task is to link detected results of an individual object in all frames to form one trajectory, which is called the ID assignment problem.

Existing MOT solutions can be categorized into two classes: 1) global optimization methods and 2) online methods. Global optimization methods [13, 59, 82, 118] minimize the total energy cost from all target trajectories. They examine all detection results of each frame and link fragmented trajectories due to occlusion. To build a more accurate energy affinity measure, a "tracklet" is defined across multiple consecutive frames and exploited to extract the spatial and temporal features of the target. Short tracklets are first generated by linking the detection results. Then, they are globally associated to build a complete trajectory of the target. Examples of global optimization methods include the graph cut [102, 103] and the flow network [87, 110, 126]. However, their performance is not satisfactory under challenging conditions such as long-term occlusion and missed detection. As there is no correctly detected bounding box for the target in both cases, the difficulty in distinguishing different objects increases along time. Moreover, in order to generate globally optimized tracks, most methods access detection results for the entire sequence beforehand, and it demands intensive computation for processing video data with iterative association. As a result, the global optimization methods are not suitable for real-time applications.

In contrast, online MOT methods are designed for real-time applications. Online MOT solutions have been studied in [6, 12, 91, 96]. The trajectory of each target is constructed frame by frame fashion, where the location and identity of one target are determined by the information of the current frame without accessing future frames. Online methods often produce fragmented trajectories with an error drift problem since it is difficult to handle inaccurate detection (or even missed detection) of occluded objects. The most challenging task in online MOT is to find an appropriate target model that correctly connects detection results of the current frame to tracks obtained from previous frames.

It is intuitive to apply the single object tracker (SOT) to the MOT problem. An online SOT can be trained and updated during the tracking process to distinguish a target from its background. Most of the state-of-the-art SOTs are built upon the convolutional neural network (CNN) architecture. They use the spatial information of the target to predict its location in the next frame, and formulate it as an end-to-end optimization problem. However, the performance is usually not satisfactory if the SOT solution is directly applied to the MOT problem. The reason is that the MOT environment is much more complicated. There exist occlusions and interactions between multiple targets, and it is challenging for a single object tracker to assign a proper identity to each target without confusion. If the identity of a target changes after occlusion/interaction, which is called the ID switch error, the error will propagate into all following frames. Thus, the design of a powerful target representation model to deal with error drift and ID switch lies in the center of the MOT problem.

To address the above-mentioned issues, we borrow ideas from human visual tracking experience and propose two target representation models in a dynamic MOT environment. If there is no occlusion for a target, we can rely on spatial-temporal consistency of the target for an incremental model update. Human eyes follow the target along the time (consecutive frames) and the brain incrementally update the gradual change of the target by comparing its appearance against the ones stored in the past. If a target is occluded, one can conduct target re-detection in the neighborhood of its original location and use the target appearance before occlusion as the reference. Once the target is recaptured after occlusion, one can initialize the tracking system with the newly detected target location and appearance, which is called the refresh update. Furthermore, we design an enhanced ID association scheme to compensate errors caused by the SOT tracker by exploiting multi-level features of the target. This is needed since the CNN tracker heavily relies on the spatial information. However, targets are sometimes small and similar, and an SOT tracker can be confused to make wrong ID association. Thus, we propose to integrate the appearance, motion and interaction cues of targets to resolve this ID switch problem.

The contributions of this work are summarized below. First, an online MOT method using multiple CNN-based SOT trackers is proposed, where each target is associated with one unique multi-domain network (MDNet) tracker [85]. It can add (Target-In) and remove (Target-Out) target trackers adaptively. Second, we present two online model update schemes: 1) the incremental update and 2) the refresh update. They work together to provide a powerful yet efficient dynamic target model in a complicated MOT environment. Third, multiple target cues are integrated and exploited to confirm the correct ID for each target.

The rest of this work is organized as follows. Sec. 5.2 offers a brief review of related work. The online MOT method is proposed in Sec. 5.3. Quantitative evaluation and experimental results are shown in Sec. 5.4. Finally, concluding remarks are given in Sec. 5.5.

5.2 Related Work

Global optimization methods. With the advancement of object detection techniques [30, 34], tracking-by-detection becomes popular for multiple objects tracking. In order to find the trajectory of each target from detection results in all frames, data association is an essential task. It is usually conducted in a discrete space using the linear programming or graph-based methods. Various optimization algorithms such as the network flow [87, 126], the continuous energy minimization [82], the max weight independent set [13], the k-partite graph [23, 121] and the subgraph multi-cut [102, 103] have been proposed. Several energy cues were introduced and optimized using the standard conjugate gradient method in [82]. In [23], each target trajectory is generated one by one in the optimization process from the best clique to the next. After finding the best one in each iteration, the corresponding detections are removed from the system. All above methods heavily rely on the detection performance. If the detection is inaccurate or missed, it is difficult for them to recover the correct target location.

Online methods. Several online MOT methods [6, 12, 91, 111] have been proposed recently to tackle with the practical real-time tracking applications. Under the "online" requirement, the ID association problem is more challenging since there are occlusions and interactions among objects. The focus has been on developing an online matching model that has an accurate feature representation so as to associate the current target location with previously detected trajectory. The part-based feature tracking was exploited in [91] to handle partial occlusion. The recurrent neural networks (RNNs) were used in [81] and [88] to manage the spatial and temporal consistency of different targets. However, the tracking capability of these methods is still limited for long-term occlusion.

Feature representations. One key component in global optimization and online methods is to define the affinity measure between two target regions. This is highly related to feature representations. Low-level features such as the color histogram and the histogram of gradients (HoG) were exploited and corner features were tracked in [9] to obtain a motion model between detection results. Supervoxels were used as the input for tracking and matching in [16, 80]. In [80], supervoxel labeling was formulated as the inference of the conditional random field (CRF) while targets were modeled as volumetric ”tubes” in a video sequence. The scene context information is exploited to link bounding boxes to form trajectories in [73].

Single object tracker in MOT. Attempts have been made in applying the SOT technique to the solution of the MOT problem [12, 116, 117, 119, 127]. Target specific classifiers were adopted to compute the similarity for data association based on particle filtering in [12]. An ensemble method was used to find the optimal candidate from detections and the tracking candidate pool in [117]. The CNN-based SOT with a spatial-temporal attention mechanism was developed in [18]. However, since its model update strategy is not good enough to train an accurate target-specific detector, the performance of the SOT tracker is not satisfactory.

In this work, we investigate a better target representation along with enhanced model update strategies so as to guarantee an accurate model to build a target-specific tracker and provide meaningful features for ID association as detailed in the next section.

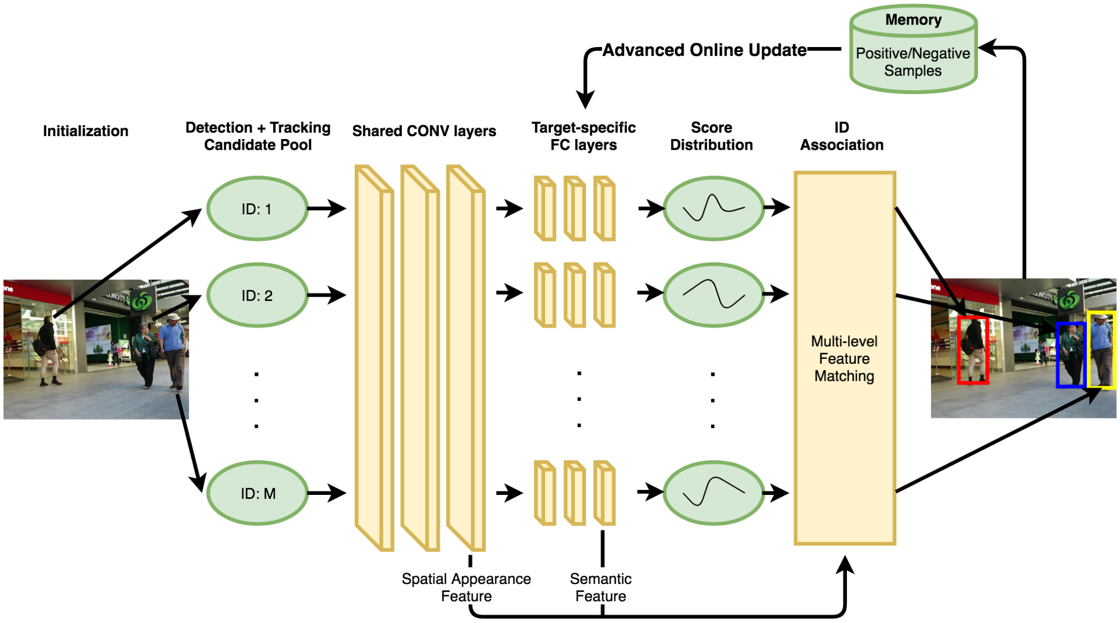

Memory
Positive/Negative Samples
Advanced Online Update
Initialization
Detection + Tracking Candidate Pool
Shared CONV layers
Target-specific FC layers
Score Distribution
ID Association
ID: 1
ID: 2
ID: M
Multi-level Feature Matching
Spatial Appearance Feature
Semantic Feature

Figure 5.1: The system flowchart of the proposed online MOT method. When the Target-In condition is met, each target is initialized with a specific tracker. In the beginning stage, trackers are working independently through the shared CONV layers and target-specific FC layers. In the later stage, the system integrates the prediction score of each tracker and multiple feature cues to assign a proper ID to each target. The online model update module manages the way to select positive and negative samples for finetuning and training the online tracker.

5.3 Proposed MOT Method

5.3.1 System Overview

An overview of the proposed MOT method is shown in Fig. 5.1. First, the system uses a Target-In condition to determine whether to initialize a target-specific branch of a CNN tracker for one object. After initialization, the system starts to track each initialized target by processing its candidates through the shared CONV layers and the target-specific FC layers. Then, by combining the score distribution information and the multiple feature cues from the CONV layers and the FC layers, it assigns an ID to each tracked target. After the location of each target in the current frame is determined, its positive and negative samples are extracted and stored in the memory as references for future model update. When the Target-Out condition is met, a target tracker is removed. The system keeps tracking remaining targets up to the end of sequence.

5.3.2 MOT CNN Tracking

In the tracking problem, the location of the m^{th} target in frame t is denoted by a bounding box

$$\mathbf{x}_m^t = (x, y, w, h)_m^t, \quad m \in \{1, \cdots, M\},$$

where M is the total number of targets in the whole video sequence, and (x, y), w and h represent the coordinates of the top-left corner, the width and the height of the bounding box, respectively. The start and terminating time instances of the m^{th} target trajectory are denoted by T_m^S and T_m^T, respectively. These two parameters are determined by the Target-In and the Target-Out conditions as described in Sec. 5.3.5.

The CNN-based tracker uses the shared CONV layers and a target-specific FC layer to track each target. The network used here is the multi-domain network (MDNet) tracker [85]. It is a small network consisting of three CONV layers and three FC layers. It is pre-trained by two commonly-used SOT benchmark datasets: the VOT dataset [54] and the OTB dataset [112]. The original MDNet was designed for the SOT problem, and it is extended to the MOT environment in this work. The shared CNN layers are fixed during the MOT tracking while a unique FC branch is assigned to each target. The system conducts online update on the FC branch so as to capture the appearance change of the target along time.

To determine the location of the m^{th} target in frame t after initialization, we search it in the candidate pool of bounding boxes:

$$C_m^t = D_m^t \bigcup (\mathbf{x}_m^{t,n})_{n=1:N_i}$$

The set, D_m^t denotes a set of detected bounding boxes in frame t that meets the following criterion. We define an enlarged window of the bounding box of the target in frame $t-1$ that contains two more pixels at each of the left/right boundaries and one more pixel at each of the top/bottom boundaries. All detected bounding boxes in frame t that overlap with this enlarged window are members in D_m^t. The spatial continuity of the target is preserved by set D_m^t.

The second set, $(\mathbf{x}_m^{t,n})$ consists of candidates generated by the tracker, where $n \in \{1, \cdots, N_i\}$ and N_i is the total number of candidates specified by the tracker. These candidates are selected using the Gaussian distribution

$$\mathbb{N}(\mathbf{x}_m^{t-1} + \mathbf{v}_m^{t-1}, \Sigma),$$

where $\mathbf{v}_m^{t-1} = \mathbf{x}_m^{t-1} - \mathbf{x}_m^{t-2}$ is the motion vector in the previous frame and $\Sigma = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_w^2, \sigma_h^2)$ is a diagonal covariance matrix. The simple linear velocity model is adopted since we focus on pedestrians with regular motion. The four variance values in Σ are evaluated independently using a Gaussian fitting process based on candidate samples in previous frames.

To estimate the target location in frame t, all candidates in pool C_m^t are evaluated by processing them through the CONV layers and the corresponding target-specific FC layers. The optimal target location is obtained by choosing the candidate with the maximum confidence score f:

$$\hat{\mathbf{x}}_m^t = \arg \max_{\mathbf{x}_m^t \in \mathbf{C}_m^t} f(\mathbf{x}_m^t, \mathbf{w}(\mathbf{fc})_m^{t-1}, \mathbf{w}(\mathbf{conv})),$$

where $\mathbf{w}(\mathbf{fc})_m^{t-1}$ denotes the network parameters of the FC layers for target m in frame $t-1$ and $\mathbf{w}(\mathbf{conv})$ are parameters of the CONV layers. After determining the location of the target in the current frame, the system checks whether the tracker model should be updated as explained below.

5.3.3 Model Update

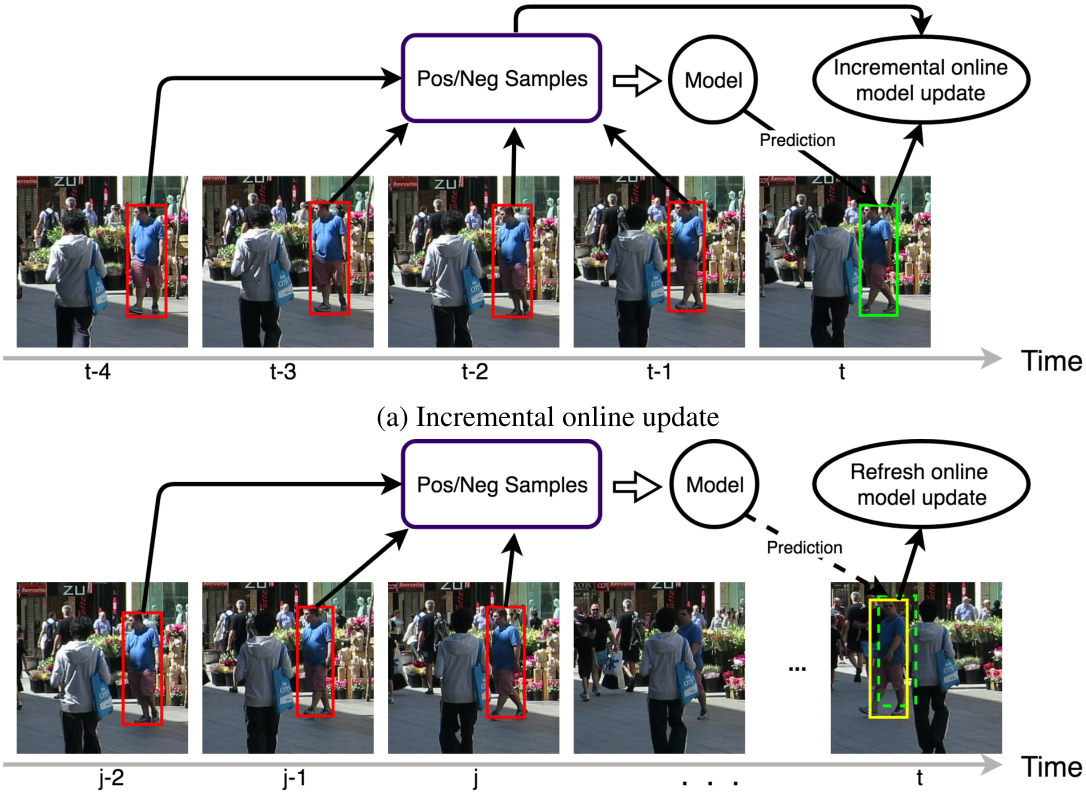
We propose two model update modes in the proposed solution. They are: 1) the incremental update and 2) the refresh update.

Incremental update

The incremental update is used to handle cases where there is no occlusion/interaction between targets. The system records the gradual appearance change of a target based on its past appearance samples as shown in Fig. 5.2(a). To track a target in each frame, the system predicts its location as indicated by the red bounding box and generates positive and negative samples around it. The score of each sample is calculated based on the intersection-over-union (IoU) ratio with respect to the prediction location. The network performs incremental update in the following two situations.

- **Tracking failure.** The system records the frame indices when the target is successfully tracked. If the confidence score of a tracked bounding box is lower than a threshold, the information in the recorded frames is retrieved to update the tracker. For example, the target location of the previously tracked frame is set to that of the current frame.

- **Periodic update.** If the tracking process goes smoothly without any failure, tracking errors can still accumulate after a number of frames. To avoid error accumulation, we set an update frequency of 20 frames to finetune the network in the experiment.



(a) Incremental online update

(b) Aggressive online update

Figure 5.2: Illustration of (a) the incremental update and (b) the refresh update, where red bounding boxes in the figures are previously tracked locations. In the incremental update, the model trained from previous positive/negative samples is used to determine the target location in current frame t, which is indicated by the green bounding box in frame t as shown in (a). If the prediction based on the incremental update is not accurate in frame t as indicated by the dashed green bounding box, the closest bounding box obtained by the detector (in yellow) is used instead in the refresh update.

With stored positive/negative samples, target-specific FC layers of the network are updated based on the sample importance. We assign the importance score to a sample with two criteria.

- **Confidence score.** The confidence score of target m in frame t, f_m^t, measures tracking accuracy of the optimal target location. If its IoU score is higher, the sample is more important.

- **Temporal relevance score.** Since the more recent appearance is more relevant, we define the temporal relevance score as

$$TI(t) = e^{\frac{2t}{T_{update}}}.$$

Then, for target m in frame t, we use the information in the past T_{update} frames to update the network with the following loss function:

$$L_m^t = \sum_{j=t}^{t-T_{update}} f_m^j \times TI(j - t + T_{update}) \times l(j),$$

where $l(j)$ is the loss of all training samples in frame j.

Refresh update

The refresh update mode is used when there are occlusions and interactions between targets. An example is illustrated in Fig. 5.2(b). Before occlusion occurs, the system collects positive and negative samples from successfully tracked frames and conducts the incremental update. When the target is occluded, it cannot find good candidates. When the target appears again, the tracker cannot perform well due to spatial and temporal discontinuities of the underlying target. Although the tracker might find a rough target location as indicated by the dashed green bounding box, its accuracy is not high and the error tends to propagate to future frames.

Being different from the SOT problem, the refresh update is feasible because detection results are available in each frame. The detection results indicate all pedestrians in the current frame without assigning an ID to existing trajectories. We may use the previous network tracker to assign an ID to each detected target. The predicted result marked by the dashed green bounding box is the optimal target location selected from the candidate pool, C_m^t, by the network. The result bounded by the yellow box is one of the detection results of the current frame that is closest to the predicted target location (i.e., the dashed green bounding box). As we can see from this example, the detected location is more accurate than the predicted one. Thus, it is chosen to be the final target location.

The refresh update mode is detailed below.

- Step 1. Use the previous tracker model to evaluate the current candidate pool C_m^t. The temporary target location is

$$\widetilde{\mathbf{x}}_m^t = \arg \max_{\mathbf{x}_m^t \in \mathbf{C}_m^t} f(\mathbf{x}_m^t, \mathbf{w}(\mathbf{fc})_m^{t-1}, \mathbf{w}(\mathbf{conv})).$$

- Step 2. Calculate the IoU score between the temporary target location $\widetilde{\mathbf{x}}_m^t$ and all detection results in D_m^t. Let

$$d^* = \arg \max_{\mathbf{d} \in \mathbf{D}_m^t} \text{IoU}(d, \widetilde{\mathbf{x}}_m^t),$$

the final target location is set to

$$\hat{\mathbf{x}}_m^t = \begin{cases} d^* & \text{if IoU}(d^*, \widetilde{\mathbf{x}}_m^t) > 0.8; \\ \widetilde{\mathbf{x}}_m^t & \text{otherwise..} \end{cases}$$

- Step 3. Refresh the tracker memory by deleting all previous samples and storing positive/negative samples generated from the current frame. The FC layers of the network is retrained based on a regression model to find a tight boundary of the target.

To summarize, the tracker is periodically finetuned in the incremental update to generate a more accurate target model to alleviate the effect of inaccurate and missed detections while a new target-specific branch is initialized to track the target by leveraging the detection result in the refresh update.

5.3.4 Enhanced ID Association

The proposed online MOT method has multiple independent target-specific branches and each of them tracks one target. For pedestrian targets in a crowed street scene, they may be small and in similar appearance. Sometimes, a tracker may jump to the location of another target as shown in Fig. 5.3, where the red and the yellow bounding boxes indicate two different targets in frame $t-1$ (see the left sub-figure) while two predicted bounding boxes merge together because of the similar appearance in frame t (see the right sub-figure). This results in an ID switch error, which will propagate to future frames. In order to handle this error, we propose an enhanced ID association mechanism that adopts multiple feature cues after independent tracking. The affinity measure between predicted bounding box $\mathbf{x}_i^t$ in frame t and previously tracked bounding box $\mathbf{x}_j^{t-1}$ in frame $t-1$ is defined by considering the following four factors (see Fig. 5.4).

- **Appearance cue.** Responses from different layers of the tracker network are utilized. Features from the CONV3 layer (of dimension $512 \times 3 \times 3$) capture the visual appearance cue while features from the FC5 layer (of dimension 512) contain semantic appearance cues. The appearance feature vector of target m in frame t, denoted as $\mathbf{A}_m^t$, is formed by cascading the above two feature vectors together of dimension 512×10.

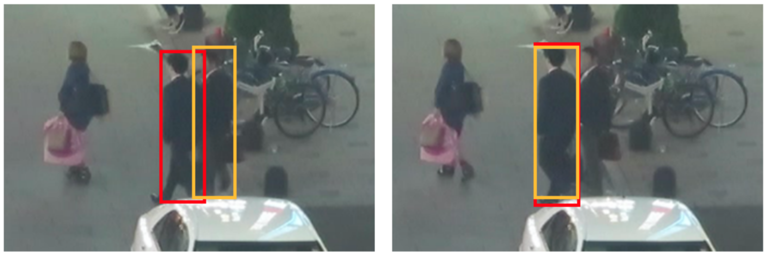


Figure 5.3: Illustration of the ”jump-merge” tracking error, where the left image shows two correct tracking boxes for two similar targets in frame $t-1$ and the right image shows the tracking result in frame t, where the yellow target region merges into the red target region resulting in an ID switch error.

- **Motion cue.** When a predicted bounding box from the tracker has an ID of target m in both frames $t-1$ and t, its motion cue is $\mathbf{v}_m^t = \mathbf{x}_m^t - \mathbf{x}_m^{t-1}$.
- **Tracking confidence score.** The tracking confidence score, denoted by f_m^t, indicates the tracking accuracy of one specific target tracker m in frame t. It is used as a weighting parameter in the final affinity measure.
- **Collision factor.** To maintain spatial consistency, we compute the spatial overlapping ratio of two different objects of IDs m and m'. If the IoU between bounding box $\mathbf{x}_m^t$ of ID m and the motion-predicted location of object of ID m' based on frame (t-1) is positive, we have to conduct the affinity measure. Mathematically, we can write the condition in form of

$$\text{IoU}(\mathbf{x}_m^t, \mathbf{x}_{m'}^{t-1} + \mathbf{v}_{m'}^{t-1}) > 0.$$

Based on the discussion, the affinity measure (AM) between the predicted bounding box $\mathbf{x}_m^t$ in frame t and previously tracked bounding box $\mathbf{x}_{m'}^{t-1}$ in frame $t-1$ is defined as

$$AM(\mathbf{x}_m^t, \mathbf{x}_{m'}^{t-1}) = \frac{\left\|\mathbf{A}_m^t - \mathbf{A}_{m'}^{t-1}\right\|_2^2 \left\|\mathbf{v}_m^t - \mathbf{v}_{m'}^{t-1}\right\|_2^2}{f_m^t f_{m'}^{t-1}\text{IoU}(\mathbf{x}_m^t, \mathbf{x}_{m'}^{t-1} + \mathbf{v}_{m'}^{t-1})}.$$

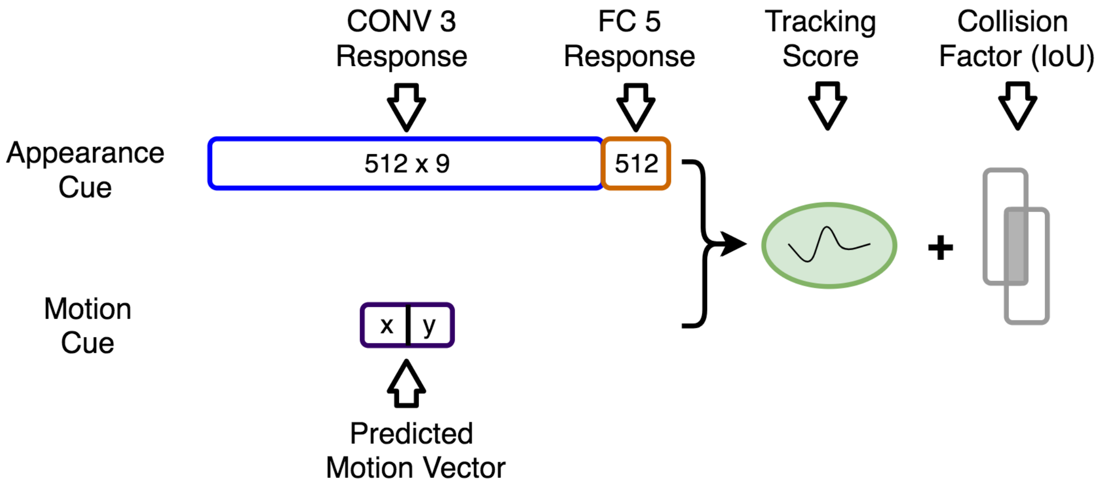


Figure 5.4: For enhanced online ID association, multiple feature cues are integrated to measure the affinity score between predicted bounding box $\mathbf{x}_i^t$ in frame t and previously tracked bounding box $\mathbf{x}_j^{t-1}$ in frame $t-1$. The appearance cue is extracted from the responses of different network layers. The motion cue is estimated from the previous target location. The final affinity measure is combined with the tracking confidence score and the collision factor between bounding boxes.

This affinity measure considers both appearance similarity and physical movement. For each tracked bounding box in current frame t, we attempt to find the most similar one in previous trajectories and then remove the matched pair in the matching procedure in future rounds.

5.3.5 Target-In and Target-Out Criteria

To manage a target tracker, we develop target-in and target-out criteria in the proposed online MOT method. The target-in criteria (to determine T_m^S) are used to initialize a target-specific tracker. They help remove the false negatives in the tracking. The target-out criteria (to determine T_m^T) are used to deactivate a tracker to avoid false positives.

Target-In Criteria

For the SOT problem, the ground truth of a target location is available in the first frame to initialize the tracker. However, for the MOT problem, only detection results are accessible in the tracking process. Inaccurate or missed detection will result in inaccurate initilization of a tracker. Two criteria are used to introduce a new tracker.

- **Confidence score.** Generally speaking, a detected target of a high score will be initialized with a tracker. However, to incorporate different score thresholds from different detectors, a filtering method based on the score distribution is proposed to choose trustworthy detection regions. Our method chooses detections with top 10% scores to initialize trackers.

- **Temporal consistency.** The confidence score of a detected target may change abruptly along time. This phenomenon often indicates a false positive, which should be eliminated in the process of initializing a new tracker. Based on the adopted constant velocity model, we set the evaluation window to five consecutive frames.

Target-Out Criteria

When a target is out of view, we should deactivate its tracker to avoid false positives. By observing sequences in the dataset, we treat the following two cases differently.

- Out of image boundaries. This occurs where a target moves out of image boundaries. We can conduct a simple boundary check to remove the associated tracker. If a target leaves the image frame and returns again, it will be treated as a new target.

- Into Background Occluders. There are two kinds of occluders: another moving target and a background occluder such as a building. For the latter, a target may go into a building which is located in the middle of an image. If a tracker fails to find the target in several consecutive frames and there is no detection overlapping region with the tracker, the tracker is deactivated.

5.4 Experimental Results

5.4.1 Implementation Details

The proposed online MOT method consists of multiple CNN-based single object trackers, implemented in MATLAB with the MatConvNet [106]. They have three shared CONV layers and three target-specific FC layers. The network is pretrained from two SOT benchmark datasets (i.e., the VOT dataset [54] and the OTB dataset [112]) using the multi-domain learning with the stochastic gradient descent (SGD) optimization technique. In the testing, the tracker is initialized with positive (IoU $>$ 0.8) and negative (IoU $<$ 0.2) samples generated from frame T_m^S. The numbers of positive and negative samples are $N_p = 500$ and $N_n = 5000$, respectively. Also, a bounding box regression model is trained to find a tight boundary for the target. For the incremental model update, the periodic update time is $T_{update} = 20$ frames. That is, the previous 20 frames are used to finetune network layers FC4-6 with $N_p = 50$ and $N_n = 200$ in each frame. We select 64 candidates from the tracking to form the candidate pool of each tracker. For the refresh update, we select $N_p = 200$ and $N_n = 1000$ samples to finetune the network.

5.4.2 Dataset and Evaluation

The proposed MOT method is evaluated on the MOT challenge dataset [79]. There are seven training and seven testing sequences. They contain multiple pedestrians and the detection results of all frames are provided for reference. Three different detection results are available in the dataset. They are generated using the DPM detector [29], the Fast RCNN detector [34] and the SDP detector [120]. For evaluation, we follow the CLEAR metrics [97]. Those include: the multiple objects tracking accuracy (MOTA), the multiple objects tracking precision (MOTP), the false positives (FP), the false negatives (FN), most tracked (MT), most lost (ML), the identity switch error (IDs) and the total fragments of all the trajectories (Frag).

5.4.3 Performance Comparison

We compare the proposed MOT method with several state-of-the-art methods on testing sequences of MOT2017 and MOT2016. All benchmarking methods and our method use the same public detection results for fair comparison. Our method does not need the training sequences to finetune the parameters while most of existing methods utilize the training sequences in the dataset. Table 5.1 shows the averaged performance results with respect to the MOT17 dataset with three public detectors. Among all published online methods for the MOT17 dataset evaluation, the proposed MOT system achieves the best performance in MOTA (44.9%), MOTP (78.9%), ID switches (1537 times) and Frag (3295 times). With the enhanced online update mode, our tracker gains a higher precision score since it can capture the target appearance more accurately. Moreover, it offers better feature representation for ID matching after target's interaction and occlusion. This advantage is clearly demonstrated in the columns of ”IDs” and ”Frag” in Table 5.1 even we compare its performance with those of offline methods. However, our false positives (FP) and false negatives (FN) are higher, which may be caused by different detectors. Actually, our FP and FN are low in average for more accurate detectors such as the FRCNN and the SDP. For the DPM detector, our tracker generates some false positives or misses some true positives due to inaccurate tracker initialization in

Table 5.1: MOT17 tracking performance in the test set with the averaged performance using three public detectors. In each mode (online/offline), the best performance is marked in bold text.

MOT17 - Test Set									
Tracker	Mode	MOTA	MOTP	MT	ML	FP	FN	IDs	Frag
IOU17 [11]	Offline	45.5	76.9	15.7%	40.5%	**19993**	281643	5988	7404
EDMT17 [15]	Offline	50.0	77.3	**21.6%**	36.3%	32279	**247297**	2264	3260
MHT_DAM [53]	Offline	50.7	**77.5**	20.8%	36.9%	22875	252889	2314	**2865**
jCC [51]	Offline	51.2	75.9	20.7%	37.4%	24986	248328	**1851**	2991
FWT [40]	Offline	**51.3**	77.0	21.4%	**35.2%**	24101	247921	2648	4279
GM_PHD [26]	**Online**	36.2	76.1	4.2%	56.6%	23682	328526	8025	11972
GMPHD_KCF [60]	**Online**	40.3	75.4	8.6%	**43.1%**	47056	**283923**	5734	7576
Ours	**Online**	**44.9**	**78.9**	**13.8%**	44.2%	**22085**	287267	**1537**	**3295**

Table 5.2: MOT16 tracking performance in the test set with one public DPM detector. In each mode (online/offline), the best performance is marked in bold text.

MOT16 - Test Set									
Tracker	Mode	MOTA	MOTP	MT	ML	FP	FN	IDs	Frag
QuadMOT16 [94]	Offline	44.1	76.4	14.6%	44.9%	6388	94775	745	1096
NOMT [17]	Offline	46.4	76.6	18.3%	41.4%	9753	87565	**359**	**504**
MCjoint [51]	Offline	47.1	76.3	**20.4%**	46.9%	6703	89368	370	598
NLLMPa [68]	Offline	47.6	78.5	17.0%	40.4%	**5844**	89093	629	768
LMP [104]	Offline	**48.8**	**79.0**	18.2%	**40.1%**	6654	**86245**	481	595
OVBT [7]	**Online**	38.4	75.4	7.5%	47.3%	11517	99463	1321	2140
EAMTT_pub [90]	**Online**	38.8	75.1	7.9%	49.1%	8114	102452	965	1657
oICF [52]	**Online**	43.2	74.3	11.3%	48.5	6651	96515	**381**	1404
STAM16 [19]	**Online**	46.0	74.9	14.6%	43.6%	6895	**91117**	473	1422
AMIR [88]	**Online**	**47.2**	75.8	14.0%	**41.6%**	**2681**	92856	774	1675
Ours	**Online**	44.0	**78.3**	**15.2%**	45.7%	7912	93215	560	**1212**

a wrongly detected location. As compared with offline methods, the proposed online method offers the competitive performance in MOTA, MOTP, IDs and Frag.

For the MOT16 dataset that has the DPM detector only, similar comparison can be conducted among online methods. As shown in Table 5.2, the proposed MOT method ranks the 3rd in MOTA, the 1st in MOTP, the 1st in MT, the 3rd in IDs, and the 1st in Frag. Overall, our tracker has higher FP and FN as compared with those in MOT17 since the DPM detector is not as accurate as the FRCNN or the SDP. However, if the target is assigned a tracker correctly, it can be tracked successfully for the whole sequence. Its precision is even higher than most offline methods.

5.5 Conclusion

In this work, an online MOT method using multiple CNN-based single object trackers was proposed. The most challenging problem in applying the SOT solution to online MOT is that the tracker can be easily confused by occlusion and interaction between targets, resulting in error drifting. Both incremental and refresh model updates were developed to address this problem. Furthermore, an ID association scheme was designed to avoid the ”jump-merge” error. It was shown by experimental results that our proposed method achieves high accuracy and precision with low ID switches.

Chapter 6

Conclusion and Future Work

6.1 Summary of the Research

In this dissertation, we studied the visual object tracking problem from two aspects: single object tracking (SOT) and multiple object tracking (MOT). In SOT, a traditional temporal prediction and spatial refinement (TPSR) tracking system, and a motion guided convolutional neural network tracker are proposed. Both of them share the idea of combining spatial and temporal motion cues into consideration. Later on, we extend the SOT technique into the MOT environment by introducing some new components, such as advanced model update and identity association, to train the tracker and control the error during the tracking.

In Chapter 3, TPSR tracking system is described, which consists of three cascaded modules: pre-processing (PP), temporal prediction (TP) and spatial refinement (SR). It has several unique features: (1) two temporal predictors (TM and OF) are working in a complementary fashion; (2) a powerful spatial refinement module makes the tracking more robust with respect to object shape and size variation; (3) it has a special mechanism to handle either partial or full occlusion. From extensive experiments, we can see the performance improvement from TPSR compared with existing the state-of-the-art methods and its advantages of handling different attributes, such as fast motion, deformation, scale variation, etc.

From a new trend of deep neural network, we investigated an tracker called motion guided convolutional neural network tracker based on a multi-domain learning framework in Chapter 4. We analyzed the drawbacks of original MDNet tracker and integrated a temporal motion cue into the network so that it can handle even more difficult scenarios, such as fast motion, articulated rotation motion and distractor. The network structure is modified to fully exploit the advantages of motion cue. With the help of optical flow map, the tracking accuracy is improved.

Based on the understanding of SOT, the MOT problem has been studied from an online fashion in Chapter 5. The most challenging task in MOT is the ID association and matching. In order to do a good job in this part, we need to find the accurate feature representation for the target so that the system can differentiate the identities. Therefore, the proposed online update strategy analyzes the two scenarios in the tracking using incremental learning and aggressive learning. It has been showed that our proposed framework is better to find the more accurate target location without ID switch.

6.2 Future Research Directions

To extend our research, we have the following research directions to further improve the proposed approaches.

- **More accurate temporal prediction in TPSR tracking system.** There are still limitations of the TPSR method. For example, it cannot provide satisfactory tracking performance for several challenging sequences in the benchmark dataset such as Ironman and Matrix due to the strong background noise, fast motion and poor quality of tracked objects. Further improvement of the TPSR tracking system in the area of temporal prediction will be helpful.

- **Long-term full occlusion in MGNet.** As we can see, short-term full occlusion is not a big issue with MGNet. However, long-term full occlusion is still a trouble maker. In this scenario, system is easy to be confused by keeping receiving incorrect target region information for a long time. Therefore, a long-term memory store needs to be integrated into the system.

- **Sensitivity of the boundary in partial occlusion.** Even though our proposed MGNet tracker can restore the correct target location after the partial occlusion, it is still sensitive to the boundaries of the target and occlusion object compared with original MDNet. One possible solution for this is to consider the spatial content of occlusion object in the network. If it is very different from the target region, then we need to let the system know there is a partial occlusion to keep the boundary of the target at original location, not be affected by the boundary of occlusion object.

- **Possibility of applying Recurrent Neural Network (RNN).** As we known, CNN is powerful to represent the spatial 2D feature in the image domain while the RNN is good at modeling the sequential reasoning. Considering that tracking is a dynamic process along the time, we can do more investigations on the temporal domain modeling using RNN architecture to find the correlation between targets in different frames. In this direction, CNN and RNN will be combined together in two different domains to handle the tracking problem.

- **Improvement on the speed.** When facing the real-world application, there will be always a trade-off between performance and speed. Currently, the CNN based tracking solution is not that fast enough for practical usage. Therefore, it is possible to do some future study on improving the speed of the solutions.

Bibliography

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805. IEEE, 2006.

[2] N. Alt, S. Hinterstoisser, and N. Navab. Rapid selection of reliable templates for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1355–1362. IEEE, 2010.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

[4] S. Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007.

[5] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009.

[6] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1218–1225, 2014.

[7] Y. Ban, S. Ba, X. Alameda-Pineda, and R. Horaud. Tracking multiple persons based on a variational bayesian model. In *European Conference on Computer Vision*, pages 52–67. Springer, 2016.

[8] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1830–1837. IEEE, 2012.

[9] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011.

[10] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Computer Vision–ECCV 2008*, pages 2–15. Springer, 2008.

[11] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by-detection without using image information.

[12] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1820–1833, 2011.

[13] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280. IEEE, 2011.

[14] R. Cabido, A. S. Montemayor, J. J. Pantrigo, M. Martínez-Zarzuela, and B. R. Payne. High-performance template tracking. *Journal of Visual Communication and Image Representation*, 23(2):271–286, 2012.

[15] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong. Enhancing detection model for multiple hypothesis tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–27, 2017.

[16] S. Chen, A. Fern, and S. Todorovic. Multi-object tracking via constrained sequential labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1137, 2014.

[17] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3029–3037, 2015.

[18] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. 08 2017.

[19] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. *arXiv preprint arXiv:1708.02843*, 2017.

[20] R. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, volume 35, 2005.

[21] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[22] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[23] A. Dehghan, S. Modiri Assari, and M. Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4099, 2015.

[24] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1177–1184. IEEE, 2011.

[25] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE, 2013.

[26] V. Eiselein, D. Arp, M. Pätzold, and T. Sikora. Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 325–330. IEEE, 2012.

[27] H. Fan and H. Ling. Sanet: Structure-aware network for visual tracking. *arXiv preprint arXiv:1611.06878*, 2016.

[28] J. Fan, W. Xu, Y. Wu, and Y. Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.

[29] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[30] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[31] R. B. Fisher. The pets04 surveillance ground-truth data sets. In *Proc. 6th IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–5, 2004.

[32] H. Fradi, V. Eiselein, J.-L. Dugelay, I. Keller, and T. Sikora. Spatio-temporal crowd density model in a human detection and tracking framework. *Signal Processing: Image Communication*, 31:100–111, 2015.

[33] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *Computer Vision–ECCV 2014*, pages 188–203. Springer, 2014.

[34] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[36] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, volume 1, page 6, 2006.

[37] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Computer Vision–ECCV 2008*, pages 234–247. Springer, 2008.

[38] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011.

[39] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):583–596, 2015.

[40] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn. Improvements to frank-wolfe optimization for multi-detector multi-object tracking. *arXiv preprint arXiv:1705.08314*, 2017.

[41] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 1495–1503, 2015.

[42] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, pages 597–606, 2015.

[43] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015.

[44] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. In *Computer Vision–ECCV 2014*, pages 155–171. Springer, 2014.

[45] Y. Hua, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In *Computer Vision–ECCV 2014*, pages 172–187. Springer, 2014.

[46] D. Huang and Y. Sun. Object tracking using discriminative sparse appearance model. *Signal Processing: Image Communication*, 37:1–18, 2015.

[47] Z. Ji and W. Wang. Object tracking based on local dynamic sparse model. *Journal of Visual Communication and Image Representation*, 28:44–52, 2015.

[48] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829. IEEE, 2012.

[49] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.

[50] F. Kemp. An introduction to sequential monte carlo methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(4):694–695, 2003.

[51] M. Keuper, S. Tang, Y. Zhongjie, B. Andres, T. Brox, and B. Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317*, 2016.

[52] H. Kieritz, S. Becker, W. Hübner, and M. Arens. Online multi-person tracking using integral channel features. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*, pages 122–130. IEEE, 2016.

[53] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[54] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015.

[55] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernandez, A. Lukežič, A. Dimitriev, et al. The visual object tracking vot2014 challenge results. In *Computer Vision-ECCV 2014 Workshops*, pages 191–217. Springer, 2014.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[57] C.-C. J. Kuo. Understanding convolutional neural networks with a mathematical model. *Journal of Visual Communication and Image Representation*, 41:406–413, 2016.

[58] C.-C. J. Kuo and Y. Chen. On data-driven saak transform. *arXiv preprint arXiv:1710.04176*, 2017.

[59] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1217–1224. IEEE, 2011.

[60] T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora. Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data.

[61] J. Kwon and K. M. Lee. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276. IEEE, 2010.

[62] J. Kwon, J. Roh, K. M. Lee, and L. Van Gool. Robust visual tracking with double bounding box model. In *Computer Vision–ECCV 2014*, pages 377–392. Springer, 2014.

[63] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[64] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden. Long-term tracking through failure cases. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 153–160. IEEE, 2013.

[65] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim. Multihypothesis trajectory analysis for robust visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5088–5096, 2015.

[66] C. Leistner, A. Saffari, P. M. Roth, and H. Bischof. On robustness of on-line boosting-a competitive study. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1362–1369. IEEE, 2009.

[67] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 277–284. IEEE, 2009.

[68] E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, and B. Andres. Joint graph decomposition &amp; node

labeling: Problem, algorithms, applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[69] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing*, 25(4):1834–1848, 2016.

[70] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[71] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[72] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5388–5396, 2015.

[73] E. Maggio and A. Cavallaro. Learning scene context for multiple object tracking. *IEEE Transactions on Image Processing*, 18(8):1873–1884, 2009.

[74] H. Masnadi-Shirazi, V. Mahadevan, and N. Vasconcelos. On the design of robust classifiers for computer vision. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 779–786. IEEE, 2010.

[75] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis &amp; Machine Intelligence*, (6):810–815, 2004.

[76] X. Mei and H. Ling. Robust visual tracking using &amp;# x2113; 1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443. IEEE, 2009.

[77] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient 1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264. IEEE, 2011.

[78] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. 03 2016.

[79] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[80] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5397–5406, 2015.

[81] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, pages 4225–4232, 2017.

[82] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multi-target tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72, 2014.

[83] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016.

[84] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. *arXiv preprint arXiv:1510.07945*, 2015.

[85] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

[86] F. Pernici and A. Del Bimbo. Object tracking by oversampling local features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(12):2538–2551, 2014.

[87] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[88] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 2017.

[89] A. Saffari, C. Leistner, M. Godec, and H. Bischof. Robust multi-view boosting with priors. In *Computer Vision–ECCV 2010*, pages 776–789. Springer, 2010.

[90] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro. Multi-target tracking with strong and weak detections. In *ECCV Workshops-Benchmarking Multi-Target Tracking*, volume 5, page 18, 2016.

[91] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012.

[92] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[93] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: an experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468, 2014.

[94] J. Son, M. Baek, M. Cho, and B. Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5620–5629, 2017.

[95] S. Song and J. Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *Proceedings of the IEEE international conference on computer vision*, pages 233–240, 2013.

[96] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *European Conference on Computer Vision*, pages 642–655. Springer, 2008.

[97] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan. The clear 2006 evaluation. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 1–44. Springer, 2006.

[98] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[99] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):39–51, 1998.

[100] J. S. Supancic and D. Ramanan. Self-paced learning for long-term tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2379–2386. IEEE, 2013.

[101] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[102] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[103] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pages 100–111. Springer, 2016.

[104] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.

[105] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 606–613. IEEE, 2009.

[106] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[107] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[108] D. Wang, H. Lu, and M.-H. Yang. Online object tracking with sparse prototypes. *Image Processing, IEEE Transactions on*, 22(1):314–325, 2013.

[109] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.

[110] X. Wang, B. Fan, S. Chang, Z. Wang, X. Liu, D. Tao, and T. S. Huang. Greedy batch-based minimum-cost flows for tracking multiple objects. *IEEE Transactions on Image Processing*, 26(10):4765–4776, 2017.

[111] S. C. Wong, V. Stamatescu, A. Gatt, D. Kearney, I. Lee, and M. D. McDonnell. Track everything: Limiting prior knowledge in online multi-object recognition. *IEEE Transactions on Image Processing*, 2017.

[112] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013.

[113] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[114] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1100–1107. IEEE, 2011.

[115] Z. Xie, J. Chen, T. Yao, and Y. Sun. Geometric structure-constraint tracking with confident parts. *Signal Processing: Image Communication*, 36:43–52, 2015.

[116] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1200–1207. IEEE, 2009.

[117] X. Yan, X. Wu, I. A. Kakadiaris, and S. K. Shah. To track or to detect? an ensemble framework for optimal selection. In *European Conference on Computer Vision*, pages 594–607. Springer, 2012.

[118] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1918–1925. IEEE, 2012.

[119] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *European Conference on Computer Vision*, pages 484–498. Springer, 2012.

[120] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2016.

[121] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer, 2012.

[122] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line semi-supervised multiple-instance boosting. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1879–1879. IEEE, 2010.

[123] J. Zhan, H. Wu, H. Zhang, and X. Luo. Cascaded probabilistic tracking with supervised dictionary learning. *Signal Processing: Image Communication*, 39:212–225, 2015.

[124] J. Zhang, S. Ma, and S. Sclaroff. Meem: Robust tracking via multiple experts using entropy minimization. In *Computer Vision–ECCV 2014*, pages 188–203. Springer, 2014.

[125] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *Computer Vision–ECCV 2014*, pages 127–141. Springer, 2014.

[126] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[127] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1838–1845, 2013.

[128] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multitask sparse learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2042–2049. IEEE, 2012.

[129] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845. IEEE, 2012.