THE UNIVERSITY
of ADELAIDE

Joint appearance and motion model for multi-class multi-object tracking

Chongyu Liu

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
School of Computer Science
The University of Adelaide

March 2019

Abstract

Model-free tracking is a widely-accepted approach to track an arbitrary object in a video using a single frame annotation with no further prior knowledge about the object of interest. Extending this problem to track multiple objects is really challenging because: a) the tracker is not aware of the objects' type while trying to distinguish them from background (*detection* task) , and b) The tracker needs to distinguish one object from other potentially similar objects (*data association* task) to generate stable trajectories. In order to track multiple arbitrary objects, most existing model-free tracking approaches rely on tracking each target individually by updating their appearance model independently. Therefore, in this scenario they often fail to perform well due to confusion between the appearance of similar objects, their sudden appearance changes and occlusion. To tackle this problem, we propose to use both appearance and motion models, and to learn them jointly using graphical models and deep neural networks features. We introduce an indicator variable to predict sudden appearance change and/or occlusion. When these happen, our model does not update the appearance model thus avoiding using the background and/or incorrect object to update the appearance of the object of interest mistakenly, and relies on our motion model to track. Moreover, we consider the correlation among all targets, and seek the joint optimal locations for all targets simultaneously as a graphical model inference problem. We learn the joint parameters for both appearance model and motion model in an online fashion under the framework of LaRank. Experiment results show that our method outperforms the state-of-the-art.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed: **Date**:

Publications

This thesis is based on the content of the following papers:

- **Chongyu Liu**, Rui Yao, S. H. Rezatofighi, Ian Reid and Qinfeng Shi; "Multi-Object Model-Free Tracking with Joint Appearance and Motion Inference"; *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2017.
- **Chongyu Liu**, Rui Yao, S. H. Rezatofighi, Ian Reid and Qinfeng Shi; "Model-free tracker for multiple objects using joint appearance and motion inference"; submitted to *IEEE Transactions on Image Processing (TIP)*, under review.

Acknowledgments

Taking the challenge to undertake the Ph.D. program is a truly life-changing experience for me, it is impossible to complete this journey without support, guidance and help from many people, I sincerely extend my thanks to them.

First of all, I would like to express my most grateful thanks to my principle supervisor Dr. Qinfeng (Javen) Shi for all the support and guidance he has given to me, from the rigorous academic training to the inspiring research ideas and all that throughout my Ph.D. period. Anytime when I approach him to seek advice or help, no matter it is research related or my personal affair, he spares no effort to help me. His understanding and encouragement are my guiding light during my difficult and struggling time. I cannot reach this point without these help.

My profound gratitude also goes to my co-supervisor, Dr. S. Hamid Rezatofighi. His truly dedicated mentoring and considerate research planning have given me the opportunity to advance steadily. I am particularly indebted to him for his constant help in my experiment and paper reviewing, he made every effort to spare more time to help me even during his busiest period.

I am also sincerely appreciative to Dr. Rui Yao, who worked closely with me in our research. He shared invaluable experience in academic research and skills in algorithm implementation and experiments, which helped me a lot in my research work.

I would like to thank my colleagues in the University of Adelaide, Teng Li, Qichang Hu and Bohan Zhuang, and my friend Kewei Wu, for their help in academic discussion.

Finally, but by no means least, my great thanks go to my parents-in-law for their help in taking care of my little boys, and to my parents, my wife Qiaoling, as well as my two little boys, Qinchen and Yuyao, for their unconditional love and endless support. I dedicate this thesis to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Visual object tracking is one of the fundamental tasks in a various computer vision applications, such as human-computer interactions, robotics, behavior analysis, surveillance, to name just a few. Given the initial state of object(s) of interest in the first image frame, the goal of visual tracking is to estimate the states of such object(s) from the second frame onwards. Although recent decades have witnessed the significant progress in visual object tracking, it still remains challenging, mainly due to heavy occlusion, severe deformation, sudden motion and varying illumination, etc. Visual object tracking may target at *single* or *multiple* object(s), forming the two major research domains: *single* object tracking and *multiple* object tracking.

For *single* object traking, significant progress has been made in tracking specific object (such as faces (Parkhi et al., 2014), pedestrians (Leykin and Hammoud, 2010), etc.), for which much prior information has been known. While tracking arbitrary objects is still hard, and model-free tracking (Babenko et al., 2011) is widely adopted to tackle arbitrary object tracking problem.

In the mean time, the *multiple* object tracking literature over the last decade has been rich, from the notable early works of multi-hypothesis tracker (Reid, 1979) and the joint probabilistic data association filter (Fortmann et al., 1980) to recent recursive (Vermaak et al., 2003; Okuma et al., 2004) and non-recursive (Berclaz et al., 2011; Butt and Collins, 2013; Milan et al., 2014) approaches. However that community has recently concentrated much more (though not exclusively) on tracking a set of pre-known objects or object classes, such as pedestrians. For this reason the dominant paradigm for multi-object tracking is one of "tracking-by-detection" (Milan et al., 2016), which assumes that in each frame it is possible to find all of the targets using some pre-trained detector, such as human detector in (Dalal and Triggs, 2005).

However, there has been much less work in model-free tracking for multiple arbitrary objects. We try to bridge this gap in the present thesis, in which we propose

a novel approach to track multiple arbitrary objects at the same time. We formulate the problem as one of inference over a graphical model, considering the correlation among all targets, and seek the joint optimal locations for all targets simultaneously. We learn the joint parameters for both appearance models and motion models in an online fashion by training a classifier with an online structured Support Vector Machines (SVMs) (Bordes et al., 2007) (Hare et al., 2011). While structred learning has previously been applied in visual tracking, this has been done either in the context of single-target tracking (*e.g.* Struck (Hare et al., 2011)), or for modelling joint motion of targets divorced from appearance changes (*e.g.* SPOT (Zhang and van der Maaten, 2014)), but not for both as in our framework. To ameliorate the issue of sudden appearance changes, we introduce an indicator variable to the graphical model that predicts sudden appearance change and occlusion and prevents updating of the appearance model.

Some preliminary results have been published in our conference paper (Liu et al., 2017a). This thesis is the complete version of the proposed framework.

1.2 Problem formulation

This thesis focuses on the problem of tracking multiple arbitrary objects, we address this problem by a joint learning and joint inference approach. Specifically, we treat the objects of interest as a probabilistic graphical model (PGM), and propose to build both appearance and motion models, and then jointly learn the models using structured SVM (Tsochantaridis et al., 2005) (Bordes et al., 2007). The learned model parameters are fed into the PGM to jointly infer the global optimal of target locations.

1.3 Main contributions

Our key contributions are: (1) We introduce a joint learning framework for all objects of interest (even of different classes), which naturally associates the target objects; (2) We introduce a motion model and confidence indicator into multiple-object visual tracking to improve the tracker's performance; (3) Our model – unlike some others that model target motion correlations via structured outputs – does not have restrictions on the graph structure and potential function type, so our method generalizes better to diverse tracking scenarios; (4) we establishe a dataset, consisting of 24 sequences, for multi-class multi-object tracking; and (5) Our extensive experiments show that our appearance and motion tracker outperforms peer trackers.

1.4 Thesis outline

The thesis is organized as follows.

In chapter 2, we review the previously published works in visual object tracking, and also elaborate popular datasets and evaluation metrics for different strands of tracking problems.

In chapter 3, we introduce background information, including structured output learning, probabilistic graphical models and convolutional neural networks.

In chapter 4, we present our approach to the problem of multiple arbitrary objects tracking. We will elaborate the joint appearance and motion inference algorithm.

In chapter 5, we intensively explore the aforementioned algorithm, including extending datasets for evaluation, incorporating deep CNNs feature and conducting more experiment.

In chapter 6, we will state some future direction of the task, and also conclude the thesis.

Chapter 2

Literature Review

As a fundamental task in computer vision, visual object tracking has been applied to various applications. Although much progress has been achieved in this topic in recent years, visual object tracking remains a very challenging task, as performance of a tracking algorithm is affected by a wide range of factors, e.g. appearance and illumination variation, occlusion, background clutters, etc.. Researchers has invested great effort to overcome these problems and design robust tracking algorithm. In this chapter, we will review the work, published in the past a few years, in visual object tracking.

2.1 Online and offline visual object tracking

Based on processing mode, there are two separate communities in visual object tracking, i.e. *online tracking* and *offline tracking*.

Visual object tracking is often referred in an *online* mode, that is, the video frame is available frame by frame, so based on the given initialized state of the target object in a video frame, the tracker is expected to adapt based on current and/or previous frame(s) and track the target object in the subsequent frames. It is suitable for online tasks, while in the meantime, it suffers from limitation of observed information. (Wu et al., 2013) comprehensively evaluated performance of a collection of online tracking algorithms, in which the algorithms were reviewed in terms of a few main modules : target object representation, search mechanism, model update and context information(for some methods). Especially, it is shown in (Wu et al., 2013) that *Struck* (Hare et al., 2011), which inspired our work, is outperforming most of the peer trackers in various evaluations, and also that dense sampling is beneficial to track fast moving target and structured learning is effective in dealing with occlusion. While in (Wu et al., 2013) all the evaluated tracking algorithms are implemented with traditional visual features, such as color histograms (Comaniciu et al., 2003), histograms of oriented gradients (HOG) (Dalal and Triggs, 2005) and Haar-like features (Viola

and Jones, 2004; Grabner et al., 2006), with the growing popularity of Convolutional Neural Networks (CNNs), CNNs feature is also incorporated in visual object tracking and leads to some prosperous results, like (Ma et al., 2015a; Bertinetto et al., 2016), etc..

On the other hand, all video frames are available in advance for *offline* visual object tracking. Applications of offline tracking includes event analysis in surveillance, video annotation, object based video compression, video motion capture, etc.. Compared to *online* tracking, *offline* tracking can use all information available in the video for optimization (Gu et al., 2011) and is more suitable for the interactive tracking task as it can benefit from a samll amount of user assistance (Wei et al., 2007). *Offline* tracking can theoretically achieve global optimal solution, however, delay in final results output is expected.

Our work will focus on *online* visual object tracking.

2.2 Long-term and Short-term visual object tracking

Based on the length of video sequence used for tracking, there are two categories of visual object tracking, *short-term* and *long-term* visual object tracking.

Much work (Babenko et al., 2011; Hare et al., 2011; Ma et al., 2015a; Danelljan et al., 2016, 2017) has been focusing on *short-term* visual object tracking, in which the length of the video sequence is usually less than one minute, that is, a few hundreds frames. According to the definition introduced by (Kristan et al., 2016b), the short-term tracking algorithms are not required to perform re-detection if losing the target. In addition to the effort made on short-term tracking algorithms, a few datasets and performance evaluation methodology have been developed and widely used, such as (Wu et al., 2015; Kristan et al., 2016b).

On the other hand, relatively less work (Kalal et al., 2012; Supancic and Ramanan, 2013; Hua et al., 2014; Ma et al., 2015b) focuses on *long-term* visual object tracking, which is targeting at video sequences at least a few minutes long and the target moves in and out of the view. In such case, re-detection is required when tracking failure, sole tracking or detection algorithm cannot solve the problem independently, while working simultaneously, they can benefit from each other to tackle the problem. Compared to short-term visual tracking, there are limited datasets (Moudgil and Gandhi, 2018; Valmadre et al., 2018) have been constructed.

2.3 Single object tracking

2.3.1 Introduction

Single object tracking (SOT) has been comprehensively studied in the past decades, although much progress has been achieved, it remains a challenging task due to numerous factors affecting the performance of the tracking algorithm, such as severe variation in appearance/illumination, occlusion, etc.. In the following context we will have a brief review on this topic, mainly organized in terms of modules for tracking algorithms.

Initialization Visual object tracking is always requiring an initialization of the object of interest, it may be manual or automatic. Manual initialization refers to the operation that annotate the target object with bounding box or ellipse, most of single object tracking algorithms fall in this category, such as (Hare et al., 2011; Babenko et al., 2011; Ma et al., 2015a), among others. In contrast, automatic initialization usually use background subtraction and blob or motion detection (Ka Ki Ng, 2010), or employ object detectors, like face (Faux and Luthon, 2012) or human (Fablet and Black, 2002) detectors.

In addition, tracking forms (how the target object is highlighted) are also determined in initialization, including bounding box, ellipse, contour, articular blocks, interest points, etc..

Appearance modeling Appearance modeling is a two-part module (Li et al., 2013), its first part, visual representation, focuses on how to construct robust object descriptor with different visual features, and its second part, statistical modeling, works towards building effective and efficient mathematical models for identifying the object of interest, which is mainly using statistical machine learning techniques. We will elaborate these two components in the following context.

- Visual representation

  Visual representation employs different types of visual features to describe target object, (Wang et al., 2015) has shown that visual feature is the most important part for the performance of a tracking algorithm. In the past decades, a wide range of visual features have been developed and applied to various computer vision tasks, including visual object tracking. These visual features span from traditional hand-crafted features to machine learned feature, such as covariance matrix(Austvoll and Kwolek, 2010), sparse coding(Zhang et al., 2013a;

Xie et al., 2014), gradient based features (Dalal and Triggs, 2005), Harr-like feature (Papageorgiou et al., 1998), color features (Takala and Pietikainen, 2007) and deep convolutional neural networks (CNNs) feature (Chu et al., 2017), just to name a few. While each feature has its strength and weakness, e.g. Harr-like feature performs well in descripting human face (Chen and Liu, 2007), high-level deep CNNs feature has more sematic information, but lose spatial information (Ma et al., 2015a), etc., so some research tried to apply hybrid visual features to the object tracking task, e.g. (Danelljan et al., 2016, 2017) combined deep CNNs feature, HOG feature (Dalal and Triggs, 2005) and Color Names (CN) (van de Weijer et al., 2009a), in which the former is the top ranked algorithm of VOT2016 challenge.

- Statistical modeling

  Briefly speaking, statistical modeling falls into two major categories, i.e. generative and discriminative. Generative models focus on modeling the appearance of the target object and searching for the most similar candidate in the video frames (Ross et al., 2008). In comparison, discriminative models adopt a different approach, which model the appearance of both the target object and the background, and then construct a classifier to separate the former from the latter (Babenko et al., 2011). Benefiting from advances in machine learning techniques applied to visual object tracking, such as boosting (Grabner et al., 2006), multiple-instance learning (Babenko et al., 2011), structured support vector machines (SVMs) (Hare et al., 2011) and deep learning (Held et al., 2016), to name just a few. The discriminative models, or hybrid ones, usually outperform the generative models, mainly because the pure generative models are not able to well handle complicated background, this result is supported by (Minka, 2005).

Inference Probabilistic inference or deterministic optimization methods have been adopted to estimate the state of the target object.

Typically, deterministic optimization methods pose the tracking problem as optimization of an objective function, in which the objective function is generally differentiable with respect to some parameters. Usually gradient descent methods can be used to efficiently solve the optimization problem and predict the target location (Fan et al., 2010; Sevilla-Lara and Learned-Miller, 2012). However, the aforementioned objective functions are usually nonlinear with multiple local minima, (Babenko et al., 2011) adopted dense sampling to tackle this problem, which paid higher expense on computational load, while in the meanwhile, (Hare et al., 2011) employed different optimization method to achieve the optimized result efficiently.

On the other hand, probabilistic inference methods represent states of target object as a probabilistic distribution with uncertainty. As such, the tracking problem is posed as a problem in estimating the probabilistic distribution of the states, in which the estimation is based on existing observations and achieved by some probability reasoning methods, such as Kalman filters (Weng et al., 2006) and particle filters (Ross et al., 2008; Dai and Liu, 2015). Kalman filters generally assume the system is linear and the target object states are Gaussian-distributed, while particle filters remove the assumption about the distribution, since they model the underlying distribution by a set of weighted particles. These kind of methods have been widely used as they are relatively insensitive to the local minimum and are computationally efficient.

Model update Model update refers to the strategy and frequency of updating the model. Many prior work (Grabner et al., 2006; Hare et al., 2011; Jia et al., 2012) has shown that online update of target representation plays an important role for handling appearance change in a robust object tracking. However, in the early stage, generative trackers are the research focus in this area. Effect of different template update strategies has been first compared in (Matthews et al., 2004), following which (Ross et al., 2008) proposed template update with incremental PCA, and recently (Danelljan et al., 2017) adopted sparser updating scheme, which updates samples each frame but updates the model only when sufficient change occurs, to reduce computational load and avoid model drift. Recently this component has also been studied in discriminative trackers, e.g. (Zhang et al., 2014) proposed a multi-expert restoration scheme, which is based on entropy minimization, to correct undesirable model update.

Context and fusion of trackers In object tracking task, there are many temporary, yet potentially strong, relationship between the object of interest and the context, such context is able to provide distinct visual property to help tracking the target. Various approaches have been developed to exploit the context information to assist tracking (Grabner et al., 2010; Dinh et al., 2011; Borji et al., 2012; Wang et al., 2017). (Grabner et al., 2010) proposed a method to learn temporally useful supporters to predict the target position, even when the target is fully occluded or out of the image; (Dinh et al., 2011) engaged a sequential randomized forest and local features to automatically explore the context, distracters and supporters, to construct the 'context tracker'; (Borji et al., 2012) used a quick training phase with user interaction at the beginning of the image sequence to learn background clusters along with target representations, and then determined the best fitting background cluster for each fol-

lowing frame and used the corresponding object representation for tracking; (Wang et al., 2017) modeled the target and context as linear combination of PCA and formed dictionary templates, and integrated the context information into subspace learning.

To overcome the limitation of single tracker and improve the tracking performance, researchers have developed some fusion methods, e.g. (Santner et al., 2010) proposed a method by augmenting an online learning method with complementary tracking approaches, that is, combining non-adaptive, highly adaptive and moderately adaptive elements in a cascade; (Kwon and Lee, 2011) proposed an approach sampling both the states of the target and the trackers, and the trackers sampled from predefined tracker space run in parallel and interact with each other to handle target variations, etc..

2.3.2 Public dataset and evaluation metrics

Performance evaluation of tracking algorithms is critically important, much work has been done in this area (Wu et al., 2010; Qing Wang, 2011; Salti et al., 2012; Pang and Ling, 2013; Wu et al., 2013; Smeulders et al., 2014; Wu et al., 2015; Kristan et al., 2016b). We will elaborate 2 major works, with far-reaching impact in the SOT area, in the following context.

Visual tracker benchmark

- Dataset

  Visual tracker benchmark's first version (Wu et al., 2013) is the first public datasets for the benchmark evaluation of online visual tracking algorithms, it contsists of 50 sequences (*TB-50*), and then the extended version (Wu et al., 2015) contains 100 sequences (*TB-100*). The dataset covers challenging sequences for visual tracking task. These sequences are categorized by their challenge attributes, including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, background clutters, etc..

- Evaluation metrics

  (Wu et al., 2015) and (Wu et al., 2013) both adopt 2 quantitative metrics to evaluate the tracking algorithms' performance, i.e. precision rate and success rate.

  Precision rate Precision rate is a metric to measure center location error (CLE), the average Euclidean distance between the ground truths and the center of the predicted locations. Usually the CLE of all frames in one video sequence is averaged to summarize the algorithms' performance in the corresponding

sequence. Precision rate is the percentage of frames for which the predicted bounding box location is under a predefined threshold distance to the ground truth. However, precision rate focuses on the bounding boxes locations only, and ignores their size and overlap, so the following metric "success rate" is preferred.

Success rate Bounding boxes overlap is another metric, it is defined as the ratio of the intersection of ground truth bounding box and predicted bounding box to the union of them, i.e.

$$\Phi_t = \frac{|b_t \cap b_a|}{|b_t \cup b_a|} \tag{2.1}$$

where b_a is the ground truth bounding box and b_t is the predicted bounding box, and the intersection and union are both calculated with pixels in the areas. Similar to precision rate, the success rate is also a ratio of successful frame at a predefined threshold. While normally the area under curve (AUC) of the success rate plot is used to evaluate the algorithms.

In addition to these 2 quantitative metrics, (Wu et al., 2013) introduced 2 ways of evaluation: temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE), which are engaged to analyze the algorithms' robustness to initialization.

Temporal robustness evaluation (TRE) Tracking algorithms start from one initial bounding box in a start frame, which may not be the first frame of the sequence, i.e. the algorithms is evaluated on a few segments of the entire sequence. Then the overall statistics are calculated to evaluate the algorithms.

Spatial robustness evaluation (SRE) The initial bounding box of the first frame is spatially shifted, including center shift, corner shift and scale variation. Then the tracking algorithms are evaluated under such circumstances.

Visual object tracking (VOT) challenge VOT challenge (Kristan et al., c, 2014, b, 2016a, a, 2016b) targeting at short-term single object model-free tracking, dates back to 2013, and it is held every year.

- Dataset

  Dataset of VOT challenge is keeping developing and improving, with the principle that the dataset represents various visual phenomena and also requires reasonably low time for performing the experiments(Kristan et al., c). In VOT2013 (Kristan et al., c), 16 sequences were selected from a large pool of video sequences used by other prior tracking research works, and each frame of these

sequences were manually or semi-manually labeled with challenging attribute, such as illumination change, motion change, occlusion, etc.; the aforementioned principle was followed in VOT2014 (Kristan et al., 2014), in which 25 sequences were manually selected after a few selection steps (refer to (Kristan et al., 2014) for details) and the same appearance attributes like VOT2013 were labeled for each frame; in VOT2015 (Kristan et al., b), the above principle was extended that the sequence selection was fully automated with carefully designed algorithm (refer (Kristan et al., b) for details) and the sequence pool was extended to 60; these all 60 sequence were used in the following VOT2016 (Kristan et al., 2016a), furthermore, in VOT2016 the ground truths were refined and a automatic bounding box generation approach was proposed; the dataset was updated in VOT2017 (Kristan et al., a) with 60 public dataset and another 60 *sequestered* dataset were constructed.

- Evaluation metrics

  The evaluation metrics of VOT challenge are also advancing along with the dataset. In VOT2013 and VOT2014, 2 metrics were adopted to evaluate the algorithms' performance: accuracy and robustness.

  Accuracy Same as prior works, such as (Wu et al., 2013), accuracy is defined as the overlap ratio of the intersection of ground truth bounding box and predicted bounding box to the union of them, see overlap measure (2.1). The experiment will be run a few times and the accuracy for each frame is averaged, yielding an average accuracy per frame; or the accuracy is averaged over all *valid frames* to summarize the accuracy for the video sequence. Note that the *valid frames* refer to frames starting from the $11-$th frames after initialization to avoid the "burn-in" period (Kristan et al., c).

  Robustness The robustness is measured by failure rate, counting the number of times the tracker loses the target (overlap defined in overlap measure (2.1) drops to 0) and re-initialization is required, where the re-initialization is performed 5 frames after the failure to avoid immediate correlation (Kristan et al., c). Like accuracy metric, the repeated experiments yield the average robustness per frame or over all frames of the sequence.

  In addition to these 2 metrics, another evaluation metric, i.e. expected average overlap measure, was introduced in VOT2015 and used as main performance measure since then.

  Expected average overlap (EAO) measure EAO is calculated as following: for a N_s frames long sequence, the average overlap Φ_{N_s} is summarized without

re-initialization even the tracker lose the target;

$$\Phi_{N_s} = \frac{1}{N_s} \sum_{i=1}^{N_s} \Phi_i \tag{2.2}$$

and then for a set of N_s frames long sequences, the expected average overlap $\hat{\Phi}_{N_s}$ is calculated; when the frame length N_s ranges from 1 to N_{max}, an *expected average overlap curve* is produced, and the EAO measure is the average value of this curve in the interval of typical short-term visual tracking sequence length.

2.4 Multi-object tracking

2.4.1 Introduction

Multiple object tracking (MOT) also has numerous direct applications , and also some other high-level computer vision tasks, such as activity recognition (Choi and Savarese, 2012), pose estimation (Pfister et al., 2015; Iqbal et al., 2017), etc., ground on MOT. MOT differs from SOT in that MOT is required to determine the number of targets and to maintain their identities, so besides the common challenges like SOT, MOT features a few special issues among others: 1) initialization and termination; 2) interaction among multiple objects, etc.. We will briefly review this topic in the following context.

Initialization Similar to SOT, there are two variants for MOT works, i.e. detection-based tracking and detection-free tracking, based on how the objects of interest are initialized (Luo et al., 2017). The detection-based methods require detectors for specific object types, the output object hypotheses of the detectors are then linked by the following tracking module to form trajectories. The detector provide priori information to tracker, but it is also because of the detector, the detection-based methods are restricted to track specific types of objects, e.g. pedestrian (Iqbal et al., 2017), vehicles (Ghasemi and Safabakhsh, 2012), etc., and the performance of such methods is highly relying on the performance of the detector.

On the other hand, the detection-free methods (Hu et al., 2012; Zhang and van der Maaten, 2013, 2014) require manual initialization of *fixed* number of targets, such methods do not rely on pre-trained detectors, so that they cannot handle the cases that targets move in and out of the view.

Appearance model Appearance model of MOT is also including two components, i.e. visual representation and statistical modeling, which are the same as that of SOT.

- Visual representation

  Different kinds of features are adopted to describe the objects of interest, such as HOG (Choi and Savarese, 2012; Izadinia et al., 2012), color histogram (Mitzel and Leibe), region covariance matrix (Porikli et al., 2006), which are also engaged in SOT, and some are typically used in MOT, e.g. probabilistic occupancy map (Berclaz et al., 2011), etc..

  Also, due to the nature of different features, they have their own strengths and weaknesses in visual representation. (Luo et al., 2017) has discussed that spatial relationship of the object region is overseen in color histogram, HOG describes rich shape information while is weak in occlusion and deformation, region covariance matrix features are robust with more cost on computation, etc..

- Statistical modeling

  Statistical modeling grounds on the visual representation. (Luo et al., 2017) categorized statistical modeling into two groups: single cue and multiple cues. The former refers to transforming distance into similarity or direct calculation of similarity, while the latter is hybrid and in which different cues can benefit from each other. In (Luo et al., 2017) the multiple cues models are summarized as 5 groups based on the information fusion strategy, i.e. boosting, concatenation, summation, product and cascading.

While importance of appearance model to MOT is different from it to SOT. In SOT, appearance model is the core component of the tracking algorithms, however, it is not so important to MOT algorithms, as the performance of MOT algorithms also relies much on the following factors.

Motion model Motion model is also referred as dynamic model, which describes the dynamic behavior of an object. By estimating potential location of the objects in future frame, motion model can reduce the search space and also help the appearance on discriminating similar objects. Generally, motion models are grouped as linear motion model and non-linear motion model.

- Linear motion model

  Linear motion model is the most popular motion model, which assumes a constant velocity (Breitenstein et al., 2009). Furthermore, on ground of such assumption, there are three specific details in building the linear motion model, i.e. velocity smoothness (Milan et al., 2014), position smoothness (Yang and Nevatia, 2012b) and acceleration smoothness (Kuo and Nevatia, 2011).

However, this model is not able to handle some more complicated cases, so non-linear motion model is proposed.

- Non-linear motion model

  Non-linear motion model is capable to yield better affinity between tracklets, so that it is engaged to model some more complicated dynamics, e.g. (Yang and Nevatia, 2012a) introduced non-linear motion model for the possibly freely moving targets.

Interaction model Interaction model, also known as mutual motion model, is introduced to understand interaction between multiple objects. There are two typical interaction models, i.e. *social force models* (Helbing and Molnár, 1995; Chen et al., 2018) and *crowd motion pattern models* (Hu et al., 2008).

The *social force models* suggest that the target's dynamic changes are guided by other targets and the environment, such models have been employed in many works (Pellegrini et al., 2009; Scovanner and Tappen, 2009; Qin and Shelton, 2012; Alahi et al., 2016).

The *crowd motion pattern models* are inspired by crowd analysis (Zhan et al., 2008), they are really useful in an over-crowded case. The motion pattern is learned by various methods and then applied to track the objects of interest.

Exclusion model Exclusion model grounds on the fact that solutions with collision between two or more targets should be penalized or completely excluded. (Milan et al., 2013) introduced two kinds of constraints, i.e. *detection-level exclusion* and *trajectory-level exclusion*, to restrict such cases. The *detection-level exclusion* means two detection, with a threshold distance apart from each other, in the same image frame cannot be assigned to the same target; and the *trajectory-level exclusion* stands for that penalization will be given based on the extent two trajectories overlap.

Inference Inference of MOT can be generally categorized into probabilistic methods and deterministic optimization methods, just like that of SOT.

Focusing on finding the maximum a posteriori (MAP) solution to the MOT problem, deterministic optimization methods cast inference of data association and target states as an optimization problem. Some approaches within this framework include bipartite graph matching (Shu et al., 2012; Qin and Shelton, 2012), dynamic programming (Wu et al., 2011; Tang et al., 2015), min-cost max-flow network flow (Wu et al., 2012; Lenz et al., 2015), conditional random field (Yang and Nevatia, 2012b; Milan et al., 2013), maximum-weight independent set (MWIS) (Brendel et al., 2011), etc..

Due to the nature of these approaches, they are more suitable for offline tracking as they requires observations from at least a time span (Luo et al., 2017).

On the other hand, by modeling the targets' states as distribution with uncertainty, probabilistic methods focus on estimating this distribution based on existing observations. Such estimation assumes Markov property in the targets' state sequence. Approaches within this framework are mainly filter based, such as Kalman filter (Li et al., 2010; Rodriguez et al., 2011), Extended Kalman filter (Mitzel and Leibe) and particle filter (Hu et al., 2012; Liu et al., 2012), etc..

2.4.2 Public dataset and evaluation metrics

Dataset and metrics are critically important for quantitatively evaluating the performance of MOT algorithms, including assessment of contribution of different components or parameters, as well as comparison of different methods.

- Dataset

  There are a few public dataset employed in prior MOT works, such as KITTI (Geiger et al., 2012), PETS2016 (Patino et al., 2016), Caltech Pedestrian (Dollár et al., 2009, 2012) and MOT Benchmark (Milan et al., 2016), etc.. Quantities of video in these dataset range from only 1 to more than 100 and total quantities of frame span from less than 100 to 250000. These dataset are essential in the progress of MOT works, while the scale is still small if compared to that of SOT, and most of the videos focus on pedestrians, some include cars. Dataset for multiple generic object tracking is still not available.

- Evaluation metrics

  Evaluation metrics is critical for a fair comparison between different MOT approaches, and a large number of metrics have been proposed in prior works (Wu and Nevatia, 2006; Bernardin and Stiefelhagen, 2008a; Schuhmacher et al., 2008; Li et al., 2009). As most MOT approaches are detection-based, so metrics for object detection are usually included. (Luo et al., 2017) categorized the metrics into two sets, each of which has a few subsets measuring the performance from different aspects.

  The first set is metrics for detection, including two subsets: accuracy and precision.

  Accuracy This subset includes *Recall*, *Precision* and *False Alarms per Frame (FAF)*. *Recal* is the ratio of correct detections to ground truth detections, *precision* means the ratio of correct detections to total result detections, and FAF an averaged number of false alarms per frame.

Precision This is measured by *Multiple Object Detection Precision (MODP)*, which is an averaged overlap between true positives and ground truths.

In addition, there are a few metrics for tracking, as following:

Accuracy This subset includes two measurements: *ID switches (IDs)* and *Multiple Object Tracking Accuracy (MOTA)*. The former stands for the number of times the MOT algorithm switches from its matched ground truth; and the latter is an integration of false positive rate, false negative rate and mismatch rate, resulting in an overall tracking performance measurement.

Precision Precision is usually measured by bounding box overlap or distance, and *Multiple Object Tracking Precision (MOTP)* is widely used to describe how precisely the tracker works.

Quality Based on the extent that the trajectory is recovered, the quality subset comprises three metrics: *Mostly Tracked (MT)*, *Partly Tracked (PT)* and *Mostly Lost (ML)*. In certain situation, *Fragmentation (FM)* is used to measure the quality of long and persistent tracking.

Robustness (Milan et al., 2016) found that many MOT algorithms may be overfitted to some specific dataset, and cannot generalize well to handle different settings. So to show robustness of the MOT algorithms, (Milan et al., 2016) proposed to use the standard deviation of *MOTA* across all testing videos.

Chapter 3

Background

In this chapter, we will introduce some background information of the thesis, including structured output learning, probabilistic graphical models and convolutional neural networks.

3.1 Structured output learning

3.1.1 Introduction

Supervised learning is an important area of machine learning, it aims at learning a function, based on sampled input-output pairs, that predicts the best response variable value to the input observation. As a kind of supervised learning, classification is to learn a mapping function $f : \mathcal{X} \mapsto \mathcal{Y}$ based on the given training sample of input-output pairs $\{(\mathbf{x}_i, y_i), i = 1, \ldots, l\} \subseteq \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}$. Conventionally, the possible values of the response variable are simple finite set, i.e. $\mathcal{Y} = \{1, \ldots, k\}$. However, it is not the case for many real-world applications, such as taxonomies of document, sequence alignment, etc., where elements of $\mathcal{Y}$ are *structured objects* describing configurations over independent components or state variables, e.g. sequences , parsing trees. Such problems are commonly called *structured output learning* problems, which are normally trends to be intractable to traditional multiclass approaches, such as (Weston and Watkins, 1998; Crammer and Singer, 2002), because of the exponentially large output space. More importantly, it is crucial to exploit the structure and dependency between the elements within the output space $\mathcal{Y}$. There are various types of approaches to structured output learning model in literature, one of the most popular is *structured support vector machine (SVM)*.

3.1.2 Structured support vector machine

Structured SVM is a generalized method from widely used SVM classifier, it is suitable for training a general classifier of structured output, and has been studied in

much previous work, e.g. (Tsochantaridis et al., 2005; Bakir et al., 2007; Bordes et al., 2007; Sarawagi and Gupta, 2008). In these literature, following the maximum-margin algorithm, the approach to the structured SVM problem is to learn a *discriminant function* $F : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ over the given input-output pairs, and by maximizing F over the output space, we can derive the mapping function f:

$$f(\mathbf{x}; \mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} \; F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \tag{3.1}$$

where $\mathbf{w}$ is the parameter vector, and here $\mathbf{y}$ denotes output *configuration*, which is different from the scalar in traditional multiclass SVM. With some function $\Phi(\mathbf{x}, \mathbf{y})$ mapping input-output pairs $(\mathbf{x}, \mathbf{y})$ into a suitable feature space endowed with inner product $\langle \cdot, \cdot \rangle$, the discriminant function F is assumed as linear in the form:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle \tag{3.2}$$

For each training input $\mathbf{x}_i \in \mathcal{X}, i = 1, \cdots, l$, we expect the value $F(\mathbf{x}_i, \mathbf{y}_i)$, the correct association, is higher than any other incorrect associations: $F(\mathbf{x}_i, \mathbf{y}), \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i$, so we have the constrains $\forall i = 1, \cdots, l, \forall \mathbf{y} \neq \mathbf{y}_i, \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle \geq 0$. Following the standard SVM derivation, the objective function of learning discriminant function F is

$$\begin{aligned} \min_{\mathbf{w}, \xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i \\ s.t. \quad & \forall i : \xi_i \geq 0 \\ & \forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle \geq 1 - \xi_i \end{aligned} \tag{3.3}$$

where C is a trade-off constant, ξ_i is the slack variable accounting for the potential violation of the constraints. As aforementioned, here $\mathbf{y}$ is a *configuration* thus $\mathcal{Y}$ is usually exponentially large, leading to that traditional multiclass approaches are not feasible.

(Tsochantaridis et al., 2005) proposed a clever *cutting plane* algorithm, *SVMstruct*, it ensures convergence but significantly reduces the constraints, and requires storage and computation of only a small part of gradient. (Bordes et al., 2007) proposed a stochastic learning algorithm, *LaRank*, it provides the same performance as *SVMstruct* while runs faster, as it solves the same optimization problem and also uses gradients sparingly as *SVMstruct*. More importantly, *LaRank* achieves nearly optimal test error rates after a single pass over the randomly reordered training set, so it is suitable for online algorithms. We will use *LaRank* as learning algorithm in this thesis.

3.1.3 Applications in computer vision

Structured SVM has been widely applied to various computer vision applications. (Liu et al., 2017b; Lucchi et al., 2012) employed structured SVM to learn conditional random fields for image segmentation, (Schwing et al., 2012) modeled the parameterization of the layout as the output of structured SVM for scene understanding, (Chen et al., 2011) treated the coordinates of all body parts as structured SVM for human pose estimation, and in object tracking area, (Hare et al., 2011) naturally modeled single object tracking as a structured SVM learning problem, and (Zhang and van der Maaten, 2013) took advantage of structured SVM to learn the configuration of individual object classifiers and structural constraints, just to name a few.

3.2 Probabilistic graphical models

3.2.1 Introduction

Various practical applications, including computer vision, involves a large collection of random variables (maybe exponentially as structured output space), storing, querying and manipulating large unstructured data are extremely difficult. The Probabilistic Graphic Models (PGMs) (Pearl, 1988; Lauritzen, 1996; Jordan, 2004; Bishop, 2006) provides a powerful and flexible framework to handle such data, as PGMs take advantages of graph to compactly represent distribution of variables and to decompose multivariate joint distributions into a set of local relationships of small subsets of variables. Conditional independence, derived from such local relationships, leads to efficient learning and inference algorithms.

Graphical models are a marriage between probability theory and graph theory (Jordan, 1999). A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comprise a *nodes* or *vertices* set $\mathcal{V}$ and a corresponding *edges* set $\mathcal{E}$. Each edge $(i, j) \in \mathcal{E}$ connects two distinct nodes $i, j \in \mathcal{V}$. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, each node $i \in \mathcal{V}$ can represent a (group of) random variable(s) $x_i \in \mathcal{X}_i$, and the edges represent probabilistic relationship between these variables. The joint probability distribution $P(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$ and $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_{|\mathcal{V}|}$, is then determined by the corresponding edges. In the following sections, we will introduce three major categories of PGMs: directed graphic models, undirected graphic models and factor graphic models.

3.2.2 Directed graphic models

Directed graphic models are also know as Bayesian networks (BNs), of which the edges have arrows to indicate a particular directionality from a *parent* node to *child*

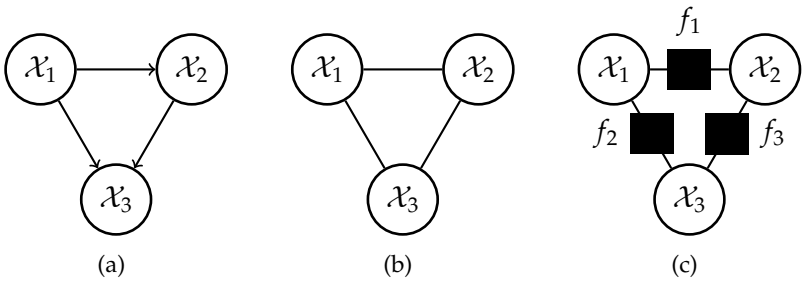

Figure 3.1: Three graph categories of a distribution over 3 random variables $\mathcal{X}_1, \mathcal{X}_2$ and $\mathcal{X}_3$. (a) Directed graph $\mathcal{G}$ depicting a causal relationship, (b) Undirected graph $\mathcal{G}$ showing Markov structure, (c) Factor graph $\mathcal{G}$ explicitly showing factorization of the graph

node, such graphs are useful for depicting causal relationships between random variables. Fig. 3.1(a) gives a simple example of directed graphs.

Directed graphic models decompose the joint distribution $P(\mathbf{x})$ into a set of conditional relationships imposed by the structure of the graph $\mathcal{G}$. To be more specific, by the product rules, joint distribution $P(x)$ can be decomposed as the product of conditional distributions of each node, where the corresponding random variable is conditioned on all of its the parents in the graph. Thus the joint distribution $P(\mathbf{x})$ is:

$$P(x_1, \cdots, x_{|\mathcal{V}|}) = \prod_{i=1}^{|\mathcal{V}|} P(x_i | Pa(x_i)) \tag{3.4}$$

where $Pa(x_i))$ denotes parent(s) of x_i. For example, the directed graph of Fig. 3.1(a) implies the following distribution $P(\mathbf{x}) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$. Such decomposition holds for all distributions and all definition of variables, if graph $\mathcal{G}$ is a *directed acyclic graph* (DAG).

3.2.3 Undirected graphical models

In undirected graphic models, the edges do not carry arrows and indicate no directional significance, such graphs are better suited to expressing correlations or constraints between random variables, instead of causal relationships in aforementioned directed graphs.

Markov Random Fields

Markov Random Fields (MRFs) is a family of undirected graphic models, in which the joint distribution $P(\mathbf{x})$ is characterized by a set of conditional independence implied by the edges. In MRFs, given its neighbors, any random variable x_i is *condition-*

ally independent to all other random variables, that is:

$$P(x_i|x_{\mathcal{V}\setminus i}) = P(x_i|x_{N_i}) \tag{3.5}$$

where x_{N_i} denotes all variables connected to x_i. This local Markov property is very important in design of efficient learning and inference algorithms of such graphic models.

Here we need to define the notion of the *clique*. A clique, c, is a set of fully connected nodes in the graph. The random variables associated with the clique can be denoted as $\mathbf{x}_c = \{x_i|i \in c\}$. According to the Hammersley-Clifford Theorem (Theorem 3.2.1), the joint distribution $P(\mathbf{x})$ can be parameterized by a product of *potential functions* defined on the cliques of the graph:

$$P(\mathbf{x}) = \frac{1}{Z}\prod_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c) \tag{3.6}$$

where $\mathcal{C}$ is the set of all cliques in $\mathcal{G}$, $\psi_c(\mathbf{x}_c)$ is the potential function over clique c and $Z = \sum_x \prod_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c)$ is a normalizer constant. For example, the joint distribution of undirected graph Fig. 3.1(b) can be written as follows:

$$P(\mathbf{x}) = \frac{1}{Z}\psi_c(x_1, x_2)\psi_c(x_2, x_3)\psi_c(x_1, x_3).$$

It is easy to find that such factorization is not unique, e.g. $P(x)$ can also be parameterized as $P(\mathbf{x}) = \psi_c(x_1, x_2, x_3)$, if we treat $\{x_1, x_2, x_3\}$ as a clique. *Maximal cliques*, the largest set of fully connected nodes in the graph, are used to get a unique factorization.

Note that there is no restriction on the choice of the potential functions to have a specific probabilistic interpretation, which is in contrast to directed graphs.

Theorem 3.2.1 ***(Hammersley-Clifford Theorem)****. Let $\mathcal{C}$ denote the set of cliques of an undirected graph $\mathcal{G}$. A probability distribution defined as a normalized product of non-negative potential functions on those cliques is then always Markov with respect to $\mathcal{G}$:*

$$P(\mathbf{x}) \propto \prod_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c) \tag{3.7}$$

Conversely, any strictly positive density ($P(\mathbf{x}) > 0$ for all x) which is Markov with respect to $\mathcal{G}$ can be represented in this factored form.

(Besag, 1974; Clifford, 1990) presented examples and further discussions about this theorem.

Pairwise Markov Random Fields

In many applications, it is useful to consider a special case of the general MRFs, the *pairwise* Markov Random Fields, in which the cliques are restricted to the pairs of nodes connected by the edges. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a pairwise MRFs expresses the joint distribution as a product of potential functions defined on that graph's edges and nodes, respectively:

$$P(\mathbf{x}) \propto \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{3.8}$$

Factor graphs

In a factor graph, besides a node for every variable in the distribution (like directed and undirected graphs), there are also additional nodes (depicted by small squares) for each factor in the joint distribution. Each factor node is connected to the variables nodes, on which that factor depends, by undirected edges. Factor graphs achieves a explicit decomposition, which is global for both directed and undirected graphs, by introducing additional nodes for the factors themselves in addition to the nodes representing the variables (Bishop, 2006), that is:

$$P(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s) \tag{3.9}$$

where $\mathbf{x}_s$ denotes a variable subset. For example, Fig. 3.1(c) expresses a joint distribution:

$$P(\mathbf{x}) = f_1(x_1, x_2) f_2(x_1, x_3) f_3(x_2, x_3)$$

For the purposes of solving inference problems,it is often convenient to convert both directed and undirected graphs into a factor graph.

3.2.4 Applications in computer vision

Because PGMs can compactly represent distribution of all variables and the relationship among their local small subsets, it has been pervasive in many different applications of computer vision, e.g. (Zhang et al., 2011) proposed a general topology chain graph and applied the corresponding learning and inference methods into human activity recognition and image segmentation, (Liang et al., 2017) modeled human limb detection and human joint localization as a unified framework, and then designed a two-steps graphical model to capture their spatial relationship in a coarse to fine way for human pose estimation, (Khémiri et al., 2014) applied PGMs to handwriting recognition, and (Hong and Han, 2014) used an offline algorithm to train a tree-structured graphic model for visual tracking, etc.

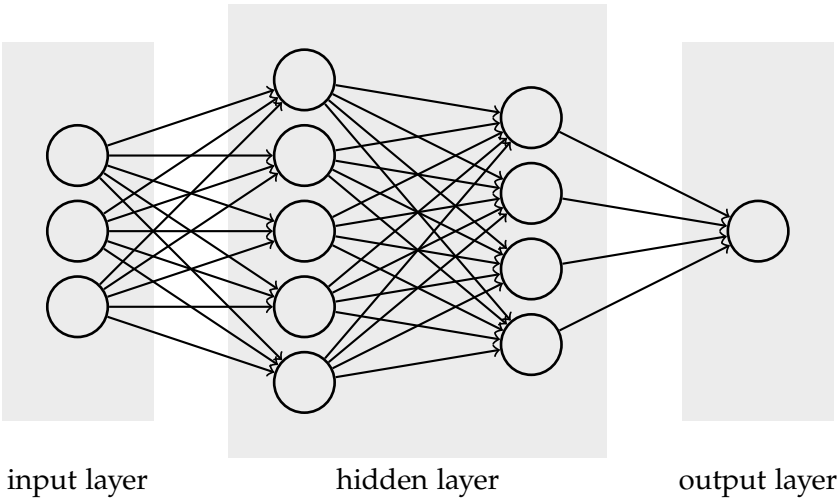

Figure 3.2: A neural network with input layer (three neurons), two hidden layers (5 neurons and 4 neurons, respectively), and output layer(one neuron)

3.3 Convolutional neural networks

3.3.1 Introduction

A conventional neural network (NN) is modeled as a collection of *neurons* connected in a acyclic graph, it consists of *input layer*, *hidden layer* and *output layer* as shown in Fig. 3.2, and such graph can define a non-linear mapping $f : \mathbb{R} \mapsto \mathbb{R}$. Each neuron in a hidden layer is fully-connected to all neurons in previous layer and receives information from them. A weighted summation of such inputs (information) from previous layer, followed by an *activation function*, generates the neuron's real-valued output, which is then propagated to all neurons in next hidden layer. By minimizing the predefined loss function, which is based on the predicted output and ground-truth, the NN learns to make prediction on unseen samples.

The number of parameters are commonly used to measure the size of conventional NNs, which is depending on the product of neuron numbers in each layer. Such characteristic of conventional NNs leads to its main drawback (massive amount of parameters to be learned) in computer vision applications, where the inputs are typically images.

Convolutional neural networks (CNNs), proposed by (Lecun et al., 1998), are derived from conventional NNs but more devised for image inputs. As CNNs benefit from the internal mechanism to significantly reduce the number of parameters, training CNNs becomes easier and it is faster to converge. Thus CNNs have been widely applied in computer vision applications. Especially after (Krizhevsky et al., 2012) showed astounding performance in a large-scale image classification challenge,

CNNs have been pervasive in all different computer vision tasks.

In this section, we will briefly introduce the architecture of CNNs and the learning algorithm, then summarize some popular variants, one of which will be applied in the thesis. In the last part we will abstract the applications of CNNs in computer vision tasks.

3.3.2 Architecture

Typically, a CNN receives an image as input, which has been subtracted mean values for each channel. This image will then be passed through the following different layers, and the layers in a CNN consist of neurons in 3 dimensions: *width*, *height* and *depth*:

- **Convolutional layer**

  The convolutional layer is a core building block of a CNN, and this layer requires heavy computational work. Parameters of the convolutional layer consist of a set of 3D learnable filters, of which the quantity is referred as the layer's depth. Each filter is spatially small in its width and height (this size is referred as *receptive field*), and its depth matches the depth of the layer's input. In the forward pass, each filter slides along the width and height of the input volume with a hyperparameter *stride*, the inner product between the filter and the input at corresponding local regions result in a 2D *feature map* after an activation function, and the feature map represents the responses of that filter at every spatial position. All produced feature maps are stacked over along the third dimension to generate the output of the convolutional layer, such output is also fed into the following layer as input. In summary, two major characteristics, local connectivity and parameter sharing, differ CNNs from conventional NNs. Local connectivity refers to that each neuron of convolutional layer only connects to a local region of the input volume, and parameter sharing means that results in a feature map are generated from the same filter. CNNs benefit from such characteristics for significantly reducing the number of parameters, compared to conventional NNs in image processing, and such reduction enables easier training and fast convergence.

- **Pooling layer**

  Pooling layer is normally located in between the successive convolutional layers in many CNNs, such as *AlexNet* (Krizhevsky et al., 2012) and *VGGNet* (Simonyan and Zisserman, 2014). The ultimate purpose of pooling operation is to control overfitting by means of reducing the spatial size of feature map, which

leads to reduction of the amount of parameters and computation of the network. Such operation always happens in a small spatial region of the input feature map, it takes mean, maximum or randomly selected value of the region to form another smaller-sized feature map.

- **Fully-connected layer**

  Fully-connected layer in CNNs is the same as that in conventional NNs, that is, all neurons in this layer have full connections to all activations in previous layer. So its activations can be computed by a matrix multiplication followed by a bias offset.

Based on these general layers, various different CNN architectures have been developed and applied to computer vision research, we will summarize a few renowned architectures in the following section.

3.3.3 Variants

In this section, we will brief some renowned CNNs architectures in computer vision research.

- **LeNet** (Lecun et al., 1998) is a pioneering CNN developed by (Lecun et al., 1998) for classifying handwritten digits. It is a 7-layer network, including 2 convolutional layers, each followed by a max pooling layer, and then fully-connected layer generating output probabilities. Constrained by computing capacity, *LeNet* requires small-sized input images.

- **AlexNet** (Krizhevsky et al., 2012) was the winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015a) in 2012, where it reached a top-5 error rate 15.3%, which significantly outperformed the runner-up by 10.8%. *AlexNet* is a revolutionary CNN model, as it showed superior performance in "feature learning" to "hand-crafted" feature, and validated CNN model in computer vision applications. This 8-layer model consists of 5 convolutional layers and 3 fully-connected layers.

- **GoogLeNet** (Szegedy et al., 2015) was the winner of ILSVRC 2014. It introduced an *Inception Module* that dramatically reduced the number of parameters in the network. Additionally, it used average pooling to replace fully-connected layers at the top of the network, achieving further reduction of a large amount of parameters. It is a 22-layer network.

- **VGGNet** (Simonyan and Zisserman, 2014) was the runner-up of ILSVRC 2014, and it showed that depth of network is critical to achieve good performance.

VGGNet is a similar model as *AlexNet* but it is "deeper", that is, *VGGNet* consists of more stacked convolutional layers. *VGGNet* has two, 16-layer and 19-layer, versions.

- **ResNet** (He et al., 2016) was the winner of ILSVRC 2015. It introduced a novel architecture *skip connections* and heavily used *batch normalization* (Ioffe and Szegedy, 2015). Thanks to these techniques, it is possible to train very deep network with 152 layers while still having lower complexity than *VGGNet*. *ResNet* has no fully-connected layers, so its number of parameters is significantly reduced.

3.3.4 Learning of CNNs

Backpropagation is originally proposed by (Rumelhart et al., 1986), and now a common method used to train NNs, as well as CNNs. The work flow is briefed as follows: the inputs are divided into *mini-batches*, which is used to approximate the gradient of predefined loss function, and then fed into the network, propagated forward through until the output layer. The cost generated by the loss function is propagated backward all the way to the input layer, while the gradients with regard to the parameters are computed using the chain rule. These gradients are applied to update the parameters with *stochastic gradient descent (SGD)* algorithm, in a way decreasing the cost.

The above mentioned learning flow is a difficult process, not only because it is time-consuming and requires huge computational resources, but also because the huge number of parameters can easily lead to overfitting. Since 2012, with help of GPU, (Krizhevsky et al., 2012) validated the feasibility of deep CNNs in computer vision applications, computational resources are not the main issue any more. In the meantime, some techniques have been developed to help training of deep CNNs, such as *dropout*, *batch normalization*, etc.. *Dropout* was originally proposed by (Hinton et al., 2012) to reduce overfitting when training NNs, it is then applied to many network architectures, such as aforementioned *AlexNet* (Krizhevsky et al., 2012) and *VGGNet* (Simonyan and Zisserman, 2014), etc.. *Batch normalization* was developed by (Ioffe and Szegedy, 2015), by normalizing layer inputs, a much higher *learning rate* can be used, which leads to a faster convergence of the network training.

3.3.5 Applications in computer vision

Since the revolutionary work *AlexNet* (Krizhevsky et al., 2012), CNNs has been pervasive in the computer vision applications, e.g. the pioneering R-CNN (Girshick et al.,

2014) applied CNN to object detection, and the following work Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017) series have dominated object detection for a few years; (Ma et al., 2015a; Bertinetto et al., 2016) and (Milan et al., 2017; Chu et al., 2017) applied pre-trained deep CNN to single object tracking and multiple object tracking application, respectively; (Pan et al., 2018) generalized traditional deep layer-by-layer convolutions to slice-by-slice convolutions within feature maps, and applied the proposed network to traffic scene understanding, etc..

Chapter 4

Multi-object model-free tracking with joint appearance and motion model

4.1 Introduction

Visual object tracking plays an important role in a wide range of computer vision applications, such as human-computer interactions, robotics, behavior analysis, surveillance, to name just a few. Despite significant progress in tracking specific object (such as faces (Parkhi et al., 2014), pedestrians (Leykin and Hammoud, 2010), etc.), for which much prior information has been known, tracking arbitrary objects is still hard, because one cannot assume there will be a well-trained detector for each object that must be tracked. Model-free tracking (Babenko et al., 2011) is a widely-accepted approach to tackle such arbitrary objects tracking problem. In model-free tracking, no other prior information about the target objects is given, except the location annotations in the first frame of video, and the target objects are expected to be tracked throughout the following video frames. Model-free tracking is a highly challenging task, as the tracker is required to distinguish not only background and foreground, but also different target objects, using very limited prior knowledge (typically just the size and location of each object in an initial frame of the sequence).

Over the last decade there has been significant progress in model-free tracking for *single* targets, with an emphasis on modelling appearance changes over time. If the tracker does not adapt the appearance model sufficiently, eventually the target will be lost because the appearance in the current image no longer matches the model well enough, but likewise inappropriate adaptation will also lead to drift and tracker failure. Key early work to address appearance changes includes subspace learning (Ross et al., 2008) in which the target is modeled via an appearance subspace, allowing linear deformations in the "appearance space".

Likewise, the multi-object tracking literature over the last decade has been rich. However that community has concentrated much more (though not exclusively) on the problem of tracking a set of pre-known objects or object classes, such as pedestrians. For this reason the dominant paradigm for multi-object tracking is one of "tracking-by-detection" (Milan et al., 2016) in which it is assumed that in each frame it is possible to find all of the targets using some pre-trained detector such as (Dalal and Triggs, 2005).

There has been much less work in the domain that would bring these two strands of work together, namely model-free tracking for multiple arbitrary objects. We address this in the following chapter, in which we propose a novel approach to track multiple arbitrary objects at the same time. We formulate the problem as one of inference over a graphical model, considering the correlation among all targets, and seek the joint optimal locations for all target simultaneously. We learn the joint parameters for both appearance models and motion models in an online fashion by training a classifier with an online structured SVM (Bordes et al., 2007) (Hare et al., 2011). While structred learning has previously been applied in visual tracking, this has been done either in the context of single-target tracking (e.g. Struck (Hare et al., 2011)), or for modelling joint motion of targets divorced from appearance changes (e.g. SPOT (Zhang and van der Maaten, 2014)), but not for both as in our framework. To ameliorate the issue of sudden appearance changes we introduce an indicator variable to the graphical model that predicts sudden appearance change and occlusion and prevents updating of the appearance model.

In this chapter, we will first summarize the proposed approach, and then elaborate the proposed algorithm, as well as the datasets and evaluation metric, followed by the experiment results comparing to the peer trackers.

4.2 Proposed approach

In the multi-object model-free tracking, given the bounding boxes of multiple objects of interest only in the first frame of the video, the tracker is expected to track all targets from the 2nd frame to the end of the video.

We treat all targets of interest in each frame as a structured output modeled by a Markov Random Field (MRF), whose parameters are learned per frame while tracking the targets. Given the parameters at each frame, the tracker localises all targets jointly via maximum a posteriori (MAP) inference maximising a score (global potential) consisting both appearance and motion. The parameters are learned in a maximum margin principle as in Struck (Hare et al., 2011), which essentially follows structured SVM (Tsochantaridis et al., 2005) and LaRank (Bordes et al., 2007).

4.2.1 Problem representation

For a video, assume there are $N \in \mathbb{Z}_{>0}$ targets to be tracked. We define a Markov random field (MRF) with graph $G = (V, E)$, where $V = \{1, \cdots, N\}$ denotes the targets, and E is the set of edges between the targets. In the t-th frame, the bounding box of target $i \in V$ is represented by $b_i^t = (c_i^t, r_i^t, w_i^t, h_i^t)$, where c_i^t, r_i^t are the column and row coordinates of the upper-left corner, and w_i^t, h_i^t are width and height, respectively. $B^t = (b_i^t, \ldots, b_{|V|}^t)$ represents all bounding boxes. Let $\mathbf{x}^t \in \mathcal{X}$ represents the frame image, and $\mathbf{y}^t = (y_1^t, \ldots, y_{|V|}^t) \in \mathcal{Y}$ represent the offsets of all targets, where $y_i^t = (\Delta c_i^t, \Delta r_i^t, \Delta w_i^t, \Delta h_i^t)$ denotes the offset of target i. The task is to learn a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, such that, given the image $\mathbf{x}^t$, the bounding boxes B^{t-1} of all targets at the $(t-1)$-th frame, and the parameter $\mathbf{w}^{t-1}$ learned at the $(t-1)$-th frame, the current bounding boxes B^t of the targets at the t-th frame can be predicted via:

$$\begin{aligned} B^t &= B^{t-1} + \mathbf{y}^{t*}, \\ \mathbf{y}^{t*} &= \arg\max_{\mathbf{y}^t \in \mathcal{Y}} f(\mathbf{x}^t, \mathbf{y}^t; \mathbf{w}^{t-1}). \end{aligned} \tag{4.1}$$

Following structured SVM (Tsochantaridis et al., 2005) and LaRank (Bordes et al., 2007), we consider the objective function $f(\mathbf{x}^t, \mathbf{y}^t; \mathbf{w}^{t-1})$ to be linear in some feature representation Φ, that is,

$$f(\mathbf{x}^t, \mathbf{y}^t; \mathbf{w}^{t-1}) = \langle \mathbf{w}^{t-1}, \Phi(\mathbf{x}^t, \mathbf{y}^t) \rangle, \tag{4.2}$$

where $\Phi(\mathbf{x}^t, \mathbf{y}^t)$ is a global feature extracted in image $\mathbf{x}^t$, and $\mathbf{w}^{t-1}$ is the parameter learned at the $(t-1)$-th frame.

4.2.2 Maximum a posterior (MAP) inference

Joint inference requires global feature, $\Phi(\mathbf{x}^t, \mathbf{y}^t)$ for all targets in the t-th frames, which consists of local features involving only one or two targets as follows,

$$\Phi(\mathbf{x}^t, \mathbf{y}^t) = \begin{pmatrix} \Big(\phi_1(\mathbf{x}^t, y_i^t)\Big)_{i \in V} \\ \Big(\phi_2(\mathbf{x}^t, y_i^t)\Big)_{i \in V} \\ \Big(\phi_3(\mathbf{x}^t, y_i^t, y_j^t)\Big)_{(i,j) \in E} \end{pmatrix}, \tag{4.3}$$

where $\phi_1(\mathbf{x}^t, \mathbf{y}_i^t)$ represents i-th target's appearance feature, $\phi_2(\mathbf{x}^t, \mathbf{y}_i^t)$ represents its motion feature, and $\phi_3(\mathbf{x}^t, y_i^t, y_j^t)$ represents the feature of edge $(i, j) \in E$.

Similarly, the parameters have 3 components

$$\mathbf{w}^{t-1} = \begin{pmatrix} (\mathbf{u}_i^{t-1})_{i\in V} \\ (p_i^{t-1})_{i\in V} \\ (\mathbf{v}_{i,j}^{t-1})_{(i,j)\in E} \end{pmatrix}. \tag{4.4}$$

corresponding to the local features.

Now f in (4.2) can be expressed as

$$\begin{aligned} f(\mathbf{x}^t, \mathbf{y}^t; \mathbf{w}^{t-1}) = \sum_{i\in V} \underbrace{\Big(\langle \mathbf{u}_i^{t-1}, \phi_1(\mathbf{x}^t, y_i^t) \rangle + \langle p_i^{t-1}, \phi_2(\mathbf{x}^t, y_i^t) \rangle \Big)}_{:=\theta_i(\mathbf{x}^t, y_i^t)} \\ + \sum_{(i,j)\in E} \underbrace{\langle \mathbf{v}_{i,j}^{t-1}, \phi_3(\mathbf{x}^t, y_i^t, y_j^t) \rangle}_{:=\theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t)}. \end{aligned} \tag{4.5}$$

Thus finding the most likely locations of all targets at each frame becomes a joint MRF inference problem:

$$\mathbf{y}^{t*} = \arg\max_{\mathbf{y}^t \in \mathcal{Y}} \sum_{i\in V} \theta_i(\mathbf{x}^t, y_i^t) + \sum_{(i,j)\in E} \theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t), \tag{4.6}$$

where $\theta_i(\mathbf{x}^t, y_i^t)$ and $\theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t)$ are node potentials and edge potentials, respectively.

Following SPOT (Zhang and van der Maaten, 2014) and Struck (Hare et al., 2011), we use fixed $\Delta w_i^t, \Delta h_i^t$ to reduce computation, though in principle they can change. Thus the inference is to find the best $(\Delta c_i^t, \Delta r_i^t)$ for all targets jointly. To reduce the computation, we define the state space for each target as follows: using the target location at the previous frame as the center, and draw an ellipse whose major axis follows the direction of the estimated velocity of the target. Every second pixel within the ellipse is considered as a feasible state for the target, which gives us about 200 to 300 states in total for each target. However, the resulting state space might still be too large for many conventional inference methods. We use a linear programming relaxation based dual message passing in (Zhang et al., 2013b), which handles a large state space, general graphs and potentials.

4.2.3 Features and Potentials

Here we elaborate the features and potentials we used, in order to overcome challenges in model-free tracking, such as heavy occlusion and poor illumination.

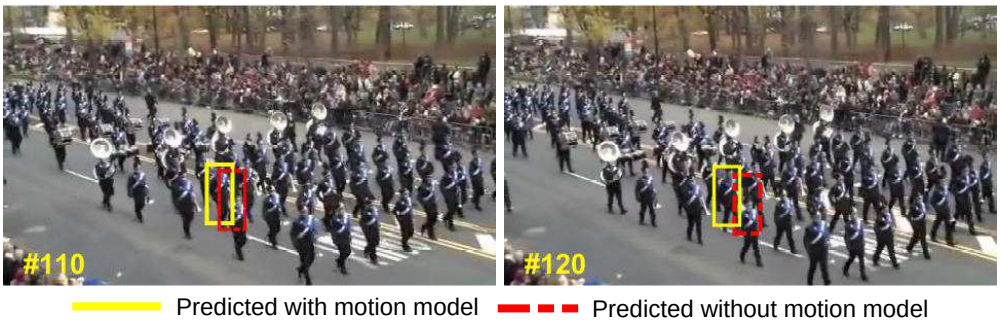

Figure 4.1: Comparison of prediction with and without motion model.

4.2.3.1 Appearance and Motion as Node Potentials

Node potential function $\theta_i(\mathbf{x}^t, y_i^t)$ consists of two parts: the appearance model $F_{A,i}^t = \langle \mathbf{u}_i^{t-1}, \phi_1(\mathbf{x}^t, y_i^t) \rangle$ and the motion model $F_{M,i}^t = \langle p_i^{t-1}, \phi_2(\mathbf{x}^t, y_i^t) \rangle$.

Appearance There are many choices for appearance feature $\phi_1(\mathbf{x}^t, y_i^t)$ as long as it represents the i-th target's appearance. The inner product $\langle \mathbf{u}_i^{t-1}, \phi_1(\mathbf{x}^t, y_i^t) \rangle$ measures the similarity between the learned parameter $\mathbf{u}_i^{t-1}$ and the candidate's appearance. This encourages the tracker to seek a location where the appearance in the current frame is close to that in the previous frame.

Motion The motion model helps in two scenarios:

- When there is a sudden appearance change (such as heavy occlusion, sudden change of illumination or sudden pose change), the appearance model's value may dramatically drop for all candidates, and the tracker often loses tracking the targets relying on appearance alone. The motion model can continue to track all targets;
- When some candidates have similar appearance, the motion model helps to distinguish them via estimating their moving directions and positions, as shown in Fig. 4.1.

The motion feature $\phi_2(\mathbf{x}^t, y_i^t)$ is defined as a Gaussian kernel below

$$\exp\Big\{ -\alpha \big[(v_{i,x}^t - v_{i,x}^{(t-1)*})^2 + (v_{i,y}^t - v_{i,y}^{(t-1)*})^2 \big] \Big\}, \tag{4.7}$$

to encourage the targets to be near the expected locations considering motion alone. Here $v_{i,x}^t$ and $v_{i,y}^t$ are estimated candidate's moving speed along X-axis and Y-axis, respectively. We simply use $v_{i,x}^{(t-1)*} = c_i^{(t-1)*} - c_i^{(t-2)*}$ and $v_{i,y}^{(t-1)*} = r_i^{(t-1)*} - r_i^{(t-2)*}$ by

assuming that for two consecutive frames, the time gap is one unit, and the target's speed slightly varies only. The scalar p_i^{t-1} essentially becomes a trade-off parameter between the appearance model and the motion model.

4.2.3.2 Edge Potentials

We use edge potentials $\theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t)$ to measure spatial relationship between two targets,

$$\begin{aligned} \theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t) &= \langle \mathbf{v}_{i,j}^{t-1}, \phi_3(\mathbf{x}^t, y_i^t, y_j^t) \rangle, \\ \phi_3(\mathbf{x}^t, y_i^t, y_j^t) &= \begin{pmatrix} e^{-\lambda(d_{i,j}^t - d_{ij}^{(t-1)*})^2} \\ e^{-\gamma(a_{i,j}^t - a_{ij}^{(t-1)*})^2} \end{pmatrix}, \end{aligned} \tag{4.8}$$

where $d_{i,j}^t$ is the Euclidean distance between target i and target j, and $a_{i,j}^t$ is the angle between edge (i,j) and X-axis. Here λ and γ are two prefixed hyper-parameters. Edge potentials encourage that two targets linked by an edge to make the smooth changes in their distance and angle.

4.2.4 Learning

Learning is also a joint process based on globle feature $\Phi(\mathbf{x}^t, \mathbf{y}^t)$, and essentially follows (Hare et al., 2011; Bordes et al., 2007) with two exceptions:

- We introduce a confidence parameter for each target to estimate how confident that the bounding box is on the true target (not the background nor other targets). If not confident, we will not update the appearance model, and the inference replies more on the motion model automatically.
- At each frame, we draw samples around the predicted targets as training data in a special way to train the appearance model to discriminate the true target against both the background and other targets.

Given training data, we solve the dual problem below as (Hare et al., 2011; Bordes et al., 2007),

$$\begin{aligned} \max_{\boldsymbol{\beta}^t} & -\sum_{j,\mathbf{y}^t} \Delta(\mathbf{y}^{t*}, \mathbf{y}^t) \boldsymbol{\beta}_{j,\mathbf{y}^t}^t \\ & -\frac{1}{2} \sum_{j,\mathbf{y}^t,k,\bar{\mathbf{y}}^t} \boldsymbol{\beta}_{j,\mathbf{y}^t}^t \boldsymbol{\beta}_{k,\bar{\mathbf{y}}^t}^t \langle \Phi(\mathbf{x}^t, \mathbf{y}^t), \Phi(\mathbf{x}^t, \bar{\mathbf{y}}^t) \rangle \\ s.t. \quad & \forall j, \mathbf{y}^t : \boldsymbol{\beta}_{j,\mathbf{y}^t}^t \leq \delta(\mathbf{y}^{(t-1)*}, \mathbf{y}^t) C \\ & \forall j : \sum_{\mathbf{y}^t} \boldsymbol{\beta}_{j,\mathbf{y}^t}^t = 0 \end{aligned} \tag{4.9}$$

where $\boldsymbol{\beta}^t_{j,\mathbf{y}^t}$ are the (simplified) dual variables, and $\delta(\mathbf{y}^{t*}, \mathbf{y}^t) = 1$, if $\mathbf{y}^{t*} = \mathbf{y}^t$ and 0 otherwise. Similar to (Blaschko and Lampert, 2008), we use the overall overlap ratio of the bounding boxes as the loss function,

$$\Delta(\mathbf{y}^{t*}, \mathbf{y}^t) = 1 - \frac{(B^t + \mathbf{y}^{t*}) \cap (B^t + \mathbf{y}^t)}{(B^t + \mathbf{y}^{t*}) \cup (B^t + \mathbf{y}^t)}. \tag{4.10}$$

By applying LaRank (Bordes et al., 2007), we maximize (4.9) to learn the dual variables, and store the support vectors. Through the dual variables and support vectors, the parameter $\mathbf{w}^t$ can be recovered,

$$\mathbf{w}^{t*} = \sum_{j,\mathbf{y}^t} \boldsymbol{\beta}^{t*}_{j,\mathbf{y}^t} \Phi(\mathbf{x}^t, \mathbf{y}^t). \tag{4.11}$$

A key step in optimising (4.9) is a SMO-style step (Platt, 1999), which requires to compute the gradient $g_j(\mathbf{y}^t)$ w.r.t. $\boldsymbol{\beta}^t_{j,\mathbf{y}^t}$ as follows,

$$g_j(\mathbf{y}^t) = \Delta(\mathbf{y}^{t*}, \mathbf{y}^t) - \sum_{k,\bar{\mathbf{y}}^t} \boldsymbol{\beta}^t_{k,\bar{\mathbf{y}}^t} \langle \Phi(\mathbf{x}^t, \mathbf{y}^t), \Phi(\mathbf{x}^t, \bar{\mathbf{y}}^t) \rangle. \tag{4.12}$$

4.2.4.1 Discriminative Sampling

Here we explain at each frame, how we draw samples to form training data set. For each target, we draw five layers ellipses centered at the ground-truth location of the target. Note that this ground-truth location is given only at the first frame, and is predicted from the second frame onward as all model free trackers do. The ellipses' major axis is in the direction of the estimated velocity. The ellipses define the state space for each target in a way slightly different from what we used in inference. We draw four types of samples as follows:

1. the ground truth locations of all targets (only once);
2. the ground truth location of one target, and randomly draw from the state spaces (within each of the ellipses) of the rest of the targets;
3. the ground truth locations of two targets which are linked by an edge, and randomly draw from the state space (within each of the ellipses) of the rest of the targets;
4. randomly draw from the state spaces (within each of the ellipses) of all targets.

Such sampling encourages the model to distinguish the true target from other targets and the background. This is particularly helpful when some targets look similar.

4.2.4.2 Confidence Parameter

When a sudden appearance change happens, we should not update the corresponding appearance parameter $\mathbf{u}_i^t$. We store the appearance similarity scores of each predicted target for the past m_k frames, a confidence parameter d_i^t is calculated as follows,

$$d_i^{t-1} = \frac{F_{A,i}^{(t-1)*}}{\sum_{j=2}^{m_k} \mu_j F_{A,i}^{(t-j)*}}. \tag{4.13}$$

where μ_j is a predefined score weight coefficient. When d_i^{t-1} is less than a certain threshold, it suggests a heavy occlusion or poor illumination occurring. We will keep $\mathbf{u}_i^t$ unchanged (i.e. not updating the appearance model). Since a small d_i^{t-1} means a low appearance score, the tracker automatically relies on the motion model more as intended.

4.3 Datasets and evaluation metric

In this section, we provide the information about the used datasets, then introduce the baselines, competitive methods, and the evaluation protocol.

Datasets We evaluate the proposed multi-object tracking algorithm on twelve challenging sequences with varied object numbers ($2-5$ objects). Sequences *basketball, red flowers, parades, shaking, skating* and *skydiving* are obtained from SPOT (Zhang and van der Maaten, 2014). The other six, *enter1cor, man kangaroo, ETH Crossing, soccer, shop2cor* and *horse racing* are public sequences, which are manually annotated with the free video (image) annotation tool *ViTBAT* (Biresaw et al., 2016). These sequences include different challenging factors for multi-object tracking: multiple interacting occlusions, appearance variations, complex deformations. Summary of each sequence such as the number of targets and the number of frames is provided in Table 4.1.

The baselines and competitive methods To demonstrate the functionality of different components of our tracker, we conduct four more baseline experiments including:

- without the edge potentials and confidence indicator, but with motion model (referred as *Ours WEWC*)
- without the edge potentials but with confidence indicator and motion model (referred as *Ours WE*)

Table 4.1: Sequences summary

Video name	Number of annotated objects	Challenges	Number of frames	Image size (Pixels)
enter1cor	2	occlusion	275	384 × 288
skating	2	bad illumination	120	640 × 360
man kangaroo	2	different kind of objects, occlusion	300	640 × 480
ETH Crossing	2	occlusion	98	640 × 480
soccer	3	very small objects	620	720 × 480
basketball	3	similar appearance, dynamic moving	195	576 × 432
shaking	3	bad illumination	344	624 × 352
shop2cor	3	occlusion	250	384 × 288
parade	4	occlusion	322	480 × 272
red flowers	4	similar appearance, dynamic moving, occlusion	1000	360 × 240
horse racing	4	similar appearance, occlusion	160	320 × 240
skydiving	5	similar appearance	1000	656 × 480

- with confidence indicator and edge potentials, but without motion model (referred as *Ours WM*)
- without the confidence indicator, but with edge potentials and motion model (referred as *Ours WC*)

To verify the performance of the proposed algorithm, we also compare it with five recent state-of-the-art trackers including: Siamese-FC (Bertinetto et al., 2016), CF (Ma et al., 2015a), KCF (Henriques et al., 2015), SPOT (Zhang and van der Maaten, 2014), and Struck (Hare et al., 2011). Siamese-FC uses fully convolutional Siamese network for tracking an object from background. CF is also a deep learning based approach, which incorporates the hierarchical CNN features into correlation filters, and track the object in a coarse-to-fine manner. KCF is one of the best correlation-filter-based trackers. Struck is a structural-learning-based tracker, which is closely related to our work and is in fact one of our baseline when we do not use motion model, joint learning and inference and our confidence parameter. Note that Siamese-FC, CF, KCF and Struck are proposed for single object tracking, we simply extend these algorithms to track multi-object by running these trackers one by one for each object. SPOT is structural-learning-based multi-object tracking algorithm, we compare our algorithm to SPOT equipped with the minimum spanning tree.

Evaluation metrics To assess the results, we employ two types of evaluation protocols: single-object tracking metrics and multi-object tracking metrics. For the former type, we use four metrics: 1) Center location error (CLE), which is computed as the average Euclidean distance between the ground-truth and the estimated center location of the tracked object. 2) VOC overlap ratio (VOR), which is defined as

$R(B_T \cap B_{GT})/R(B_T \cup B_{GT})$, where $R(B)$ measure the area of the bounding box B, B_T and B_{GT} are the tracking bounding box and the ground truth bounding box, respectively. 3) Precision plot, which measures the percentage of successfully tracked frames. Tracking on a frame is considered successful if the distance between the centers of the predicted box and the ground truth box is under a fixed threshold. 4) Success plot, which is defined as the percentage of frames where VOC overlap ratio is larger than a certain threshold (Wu et al., 2013). For multi-object tracking metrics, we follow the evaluation protocol of (Milan et al., 2016), where five of those metrics are used: Recall represents the percentage of the detected targets, IDS represents mismatch error, FAR represents the number of false alarms per frame, MT represents the number of mostly tracked targets, MOTA means multi-object tracking accuracy, and MOTP means multi-object tracking precision.

4.4 Experiments

In this section, we will elaborate implementation details of our experiments, and then present quantitative evaluation, as well as qualitative evaluation, of the proposed method.

4.4.1 Implementation Details

Similar to visual tracker (Hare et al., 2011), the proposed tracker performs object localization using a sliding-window search scheme with an adaptive search radius. Generally, for each target, we search in a 2D area with $\{(\Delta c_i, \Delta r_i)|(\Delta c_i)^2 + (\Delta r_i)^2 \leq R_i^2\}$, and sample on a polar grid. We use 5 radial and 18 angular divisions, giving 81 locations, which are used for discriminating sampling. In our implementation, the $2nd$ to $4th$ types of samples in discriminating sampling are repeated 27 times, 16 times and 81 times, respectively. As different targets may be different sizes, to acquire effective samples, we set $R_i = 1.2 \times min(w_i^t, h_i^t)$. When the motion model applies, we sample within an ellipse whose major axis aligns with the direction of the target's velocity, and its length depends R_i and target's speed. As for confidence parameter, we set up the threshold $d_i^{t-1} = 0.1$ based on the past $m_k = 5$ frames, accordingly the score coefficients μ_j are set to 0.4, 0.3, 0.2 and 0.1, respectively. Also we fix hyper-parameters $\alpha = 0.05$ for motion feature and $\lambda = 0.035$ and $\gamma = 0.025$ for edge potentials , respectively.

To perform a fair comparison with peer trackers SPOT and KCF, we also extract the same features, i.e. Histogram of oriented gradients (HOG)(Dalal and Triggs, 2005), for tracking multiple objects. For our tracker, we use fully connected graphs

in images where targets are expected to not move very fast (i.e. since maintaining distances and angles between targets in that case is better), and use simple chain structure otherwise.

4.4.2 The state space for MRF inference

As aforementioned, jointly predicting the most likely locations of all targets can be cast as an MRF inference problem. Here we elaborate how to define the state space of the inference.

Our algorithm has two variants: one without the motion model (referred to as *Ours WM* in the experiment) and one with the motion model (referred as *Ours* in the experiment). When the motion model is not used, the state space for the MRF inference is defined by a circle with radius $\bar{R}$, where $\bar{R} = 20$ (pixels) is a prefixed constant. We consider every second pixel within the circle as a feasible state to reduce the computation, which works well in practice.

When the motion model is used, we take the target location at the previous frame as the center, and draw an ellipse whose major axis follows the direction of the estimated velocity of the target. The state space is defined by an ellipse with its parameters such as semi major, semi minor and its rotation angle as follows. The semi major and the semi minor are calculated as

$$a_i^{t-1} = \bar{R} + \sqrt{(v_{i,x}^{(t-1)*})^2 + (v_{i,y}^{(t-1)*})^2}, \tag{4.14}$$

$$b_i^{t-1} = \frac{(\bar{R})^2}{a_i^{t-1}}, \tag{4.15}$$

where

$$\begin{aligned} v_{i,x}^{(t-1)*} &= c_i^{(t-1)*} - c_i^{(t-2)*}, \\ v_{i,y}^{(t-1)*} &= r_i^{(t-1)*} - r_i^{(t-2)*}. \end{aligned} \tag{4.16}$$

Here the semi minor b_i^{t-1} is calculated assuming that the ellipse keeps the same search area as the circle (without the motion model), which is found empirically helpful in our experiments.

The rotation angle is calculated as

$$\theta_i^{t-1} = -\arctan\Big(\frac{v_{i,y}^{(t-1)*}}{v_{i,x}^{(t-1)*}}\Big). \tag{4.17}$$

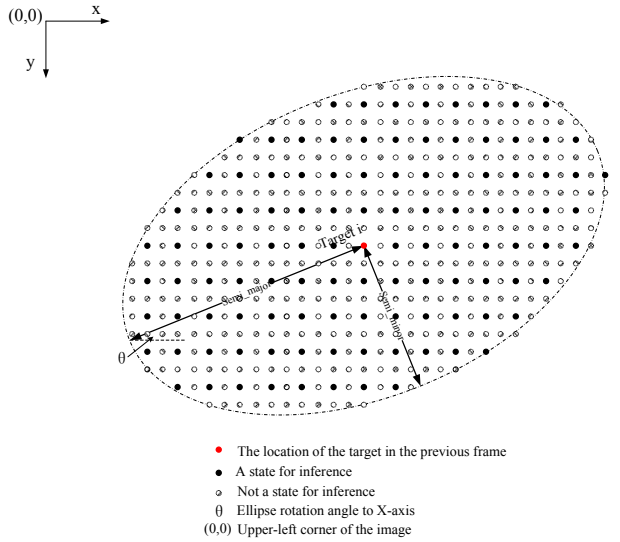

Figure 4.2: The state space for inference when considering motion

Note (4.17) is applicable only when $|v_{i,x}^{(t-1)*}| \geq 1$ pixel or $|v_{i,y}^{(t-1)*}| \geq 1$ pixel. Otherwise we consider the target's state space is defined by a circle, which is the same as the case without the motion model.

An illustration of the ellipse for the i-th target is shown in Fig.4.2 depending on the velocity of the target. Same as the case without the motion model, every second pixel within the ellipse is considered as a feasible state.

4.4.3 Sampling training data

Here we elaborate at each frame, how to sample around the targets' ground truth (or predicated) location to form training data.

When the motion model is disabled, for the i-th target at the t-th frame, the sampling space for structured SVM learning is empirically defined by 5 concentric circles with the most outer layer radius defined as

$$R_i^t = 1.2 \times \min(w_i^t, h_i^t)$$

to accommodate the fact that different targets often have different sizes in multi-object tracking scenario. We find this radius works well in our experiments.

When the motion model is enabled, we use 5 ellipses shown in Fig. 4.3 to define the candidates of the training data to be sampled.

All 5 ellipses share one center and the same rotation angle. Semi major and semi

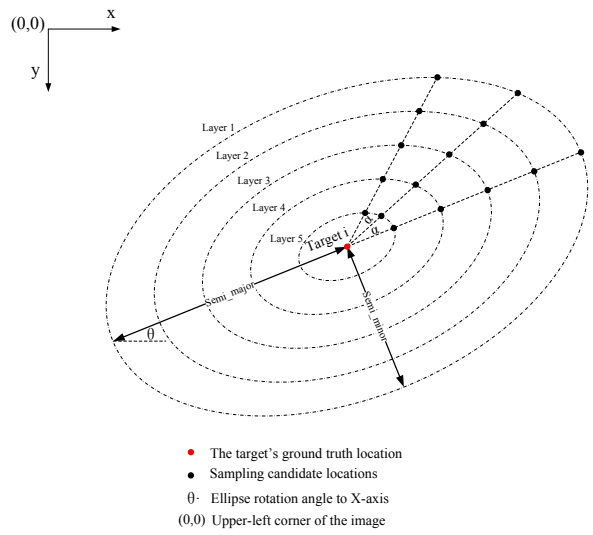

Figure 4.3: Training data sampling when considering motion

minor of the most outer ellipse (layer 1) are calculated as follows,

$$\begin{aligned} a_i^t &= R_i^t + \sqrt{(v_{i,x}^{t*})^2 + (v_{i,y}^{t*})^2}, \\ b_i^t &= \frac{(R_i^t)^2}{a_i^t} \end{aligned} \tag{4.18}$$

Semi major and semi minor of the most inner ellipse (layer 5)'s are $a_i^t/5$ and $b_i^t/5$, respectively. As for the rest 3 ellipses, their semi major and semi minor are evenly spaced between $[a_i^t/5, a_i^t]$ and $[b_i^t/5, b_i^t]$. $v_{i,x}^{t*}$ and $v_{i,y}^{t*}$ follow (4.16) and the rotation angle θ_i^t follows (4.17).

Similarly, θ_i^t is applicable only when $|v_{i,x}^{t*}| \geq 1$ pixel or $|v_{i,y}^{t*}| \geq 1$ pixel, otherwise we consider the target's sampling space is defined by 5 concentric circles, which is the same as the case without motion model.

As we can see from Fig.4.3, the sampling candidates (black dots) are evenly distributed on each ellipse, and the angle gap between two nearest samples on one ellipse is $\alpha = \frac{\pi}{9}$. This gives 81 candidates, including one ground truth (red dot).

We draw four types of samples to form the training data as stated in 4.2.4.1, such mechanism encourages the model to distinguish the true target from other targets and the background. This is particularly helpful, when some targets look similar.

Table 4.2: Quantitative evaluation of average VOR (VOC overlap ratios) and CLE (center location errors) of the proposed algorithm with 4 baselines on the reported sequences. We report the average VOR and CLE of each tracker for all objects in each sequences, and all objects in all sequences (the last row in the table). The best results are shown in bold.

Sequence	*Ours WEWC*		*Ours WE*		*Ours WM*		*Ours WC*		***Ours***	
	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR
enter1cor	5.2	0.85	4.1	**0.87**	5.2	0.85	5.2	0.85	**3.9**	0.86
skating	21.8	0.60	21.9	0.59	20.3	0.62	22.7	0.58	**19.7**	**0.64**
man kangaroo	23.2	0.68	25.9	0.68	9.1	0.81	8.9	0.81	**7.6**	**0.81**
ETH Crossing	55.4	0.64	57.0	0.64	14.1	0.79	**11.5**	**0.82**	11.6	0.82
soccer	40.2	0.42	16.7	0.59	33.0	0.43	3.5	0.68	**3.6**	**0.69**
basketball	10.6	0.75	9.9	0.76	8.7	0.78	9.6	0.76	**6.6**	**0.84**
shaking	6.4	**0.82**	**6.4**	0.82	10.7	0.77	10.6	0.77	7.2	0.80
shop2cor	8.5	0.73	8.6	0.73	7.8	**0.76**	7.9	0.74	**7.5**	0.75
parade	6.7	0.71	6.5	0.71	9.4	0.69	4.7	**0.79**	**4.6**	0.79
red flowers	17.6	0.69	19.6	0.66	11.1	0.75	11.6	0.74	**8.1**	**0.80**
horse racing	3.9	0.71	3.9	0.71	6.5	0.66	3.9	0.71	**3.3**	**0.74**
skydiving	7.3	0.77	8.3	0.76	**4.4**	**0.82**	5.3	0.80	4.7	0.81
Average	17.2	0.70	15.7	0.71	11.7	0.73	8.8	0.75	**7.4**	**0.78**

Table 4.3: The table shows baseline multi-object tracking results of baseline methods in terms of MOT performance criteria on all sequences.

Tracker	Recall	FAR	MT	MOTA	MOTP
Ours WEWC	83.4	0.59	25	66.5	**81.6**
Ours WE	84.3	0.56	26	68.4	81.4
Ours WM	86.6	0.48	29	72.9	81.5
Ours WC	89.6	0.37	29	79.2	81.0
Ours	**94.3**	**0.20**	**34**	**88.6**	81.1

4.4.4 Quantitative Evaluation

We first carry out the experiments to show the contribution of each component in our algorithm and then compare the proposed method with the aforementioned state-of-the-art methods.

Evaluation on baselines of functional components. To explore the effectiveness of different components of our tracker, we report the results of proposed algorithm with each component switched on or off in the tables. The results in Tables 4.2 and 4.3 show that each component consistently contribute in improving the total performance of the proposed method with respect to all reported metrics. For example, considering the initial baseline as "*Ours WEWC*", we can see from Table 4.3 that incorporating confidence parameter only ("*Ours WE*"), edge potential term (joint learning and inference) only ("*Ours WC*") and both terms together ("*Ours*") can respectively improve the results with respect to MOTA metric by 2.8%, 19.1% and 33.2%.

Table 4.4: Quantitative evaluation of average VOR (VOC overlap ratios) and CLE (center location errors) of the proposed algorithm with 5 state-of-the-art methods on the reported sequences. We report the average VOR and CLE of each tracker for all objects in each sequences, and all objects in all sequences (the last row in the table). The best results are shown in bold.

Sequence	Siamese-FC		CF		KCF		SPOT		Struck		***Ours***	
	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR
enter1cor	5.9	0.60	4.9	0.85	5.9	0.84	5.5	0.82	8.3	0.64	**3.9**	**0.86**
skating	83.2	0.39	57.0	**0.68**	82.8	0.50	41.0	0.51	102.5	0.43	**19.7**	0.64
man kangaroo	15.2	0.67	8.1	0.80	47.3	0.48	41.9	0.46	35.4	0.56	**7.6**	**0.81**
ETH Crossing	22.3	0.62	52.8	0.63	34.8	0.64	30.3	0.65	74.9	0.47	**11.6**	**0.82**
soccer	4.1	0.63	**3.5**	0.64	25.1	0.54	202.3	0.06	89.9	0.35	3.6	**0.69**
basketball	45.3	0.51	8.2	0.77	6.9	0.81	73.0	0.50	74.9	0.45	**6.6**	**0.84**
shaking	9.0	0.73	166.3	0.03	31.6	0.55	11.1	0.75	148.0	0.12	**7.2**	**0.80**
shop2cor	16.5	0.43	34.7	0.40	35.2	0.38	31.9	0.59	24.1	0.44	**7.5**	**0.75**
parade	30.1	0.30	21.9	0.38	21.1	0.39	25.5	0.25	46.8	0.30	**4.6**	**0.79**
red flowers	12.9	0.55	12.6	0.72	9.7	0.73	**8.1**	0.78	49.1	0.30	8.1	**0.80**
horse racing	4.4	0.65	3.8	0.70	**3.1**	0.69	7.1	0.58	23.6	0.37	3.3	**0.74**
skydiving	19.3	0.45	5.7	0.76	8.3	0.71	7.2	0.70	50.4	0.33	**4.7**	**0.81**
Average	22.3	0.54	31.6	0.61	26.0	0.60	40.4	0.55	60.6	0.40	**7.4**	**0.78**

Table 4.5: The table shows multi-object tracking results of competitive methods in terms of MOT performance criteria on all sequences.

Tracker	Recall	FAR	MT	MOTA	MOTP
Siamese-FC (Bertinetto et al., 2016)	66.9	1.18	17	33.7	69.7
CF (Ma et al., 2015a)	86.3	0.49	27	72.6	76.7
KCF (Henriques et al., 2015)	86.6	0.48	28	72.9	75.1
SPOT (Zhang and van der Maaten, 2014)	79.4	0.74	22	58.8	75.2
Struck (Hare et al., 2011)	41.6	2.09	7	-17.8	74.9
Ours	**94.3**	**0.20**	**34**	**88.6**	**81.1**

Comparison with state-of-the-art methods. Tables 4.4 and 4.5 show comparison between our proposed approach and the aforementioned competitive methods on the reported sequences using average CLE and VOR, and also MOT criteria. The proposed method significantly improves the baseline structural-learning-based tracker. Our tracker obtains almost the best performance on almost all sequences among all the compared trackers with an average CLE of 7.4 pixels and average VOR with 0.78.

As shown in Fig. 4.4, the precision plot and success plot for all datasets indicate that our method outperforms four other state-of-the-art trackers. All algorithms are compared in terms of the same initial positions in the first frame. For precision plots and success plots, the values in the legend are the average distance precision at 20 pixels and the Area Under Curve (AUC) scores, respectively. In both plots, the best method is the proposed tracker; our algorithm ("Ours") outperforms the competitors in both mean distance precision and mean AUC scores, respectively.

In Table 4.5, we compare the performance of our proposed tracker with the competitive trackers using multi-object tracking (MOT) metrics. Table 4.5 shows that our

method outperforms other methods overall. The MOTA is perhaps the most widely used metric to evaluate a multi-object tracker's performance (Bernardin and Stiefelhagen, 2008b). It takes three sources of errors (i.e. the number of false positive, of misses, and of mismatches). Note that a negative MOTA means that the number of errors made by the tracker exceeds the number of all objects. The MOTP is the average dissimilarity between all true positives and their corresponding ground truth targets. The higher the Recall, MT, MOTA and MOTP of a tracker, the better its performance. On the contrary, the lower FAR indicates a better performance. In Table 4.5, SPOT outperforms the other existing methods in our evaluation, this maybe because that SPOT takes spatial structure among objects into account, it will provide more information for tracking multi-object. By utilizing more effective node potential and edge potential, our tracker outperforms SPOT 7.8% in MOTP.

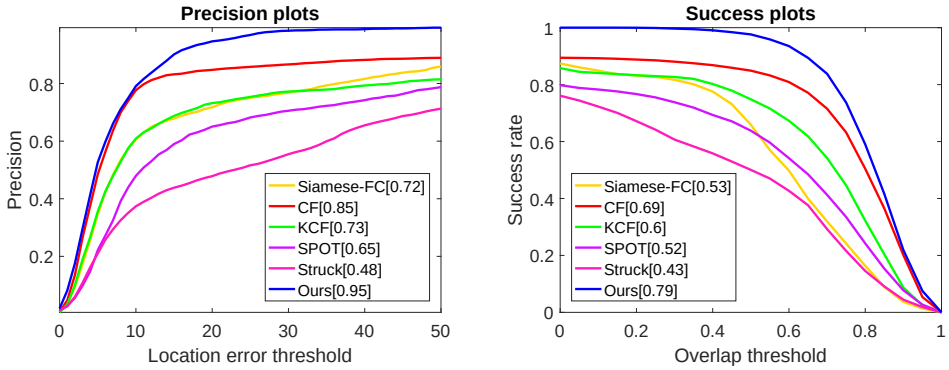

Figure 4.4: Overall distance precision plot (left) and overlap success plot (right) over all sequences. The legends show the precision scores and AUC scores for each tracker.

4.4.5 Qualitative Evaluation

We present several tracking results from the proposed tracker in different rows of Figs. 4.5. For presentation clarity, the tracking results of the other trackers are not shown in the figure. Overall, our tracker is able to produce precise localizations for the multiple objects. Note that in two of the sequences, (*basketball* and *horse racing*) shown in the first two rows, there are deformations and occlusions. In the following two rows, our tracker performs well in presence of heavy interacting occlusion between similar objects. When the background is cluttered (sequence *parade*) and the illumination is changed (sequence *shaking*), the proposed multi-object tracker still keep track of the objects because it leverages the structural information encoded in the graph edge potentials. The 7th and 8th rows also show some cases of occlusion, deformation, and scale variations. The range of these sequences and different

challenges presented in each, demonstrate that our tracker performs well in various circumstances.

Figure 4.5: Multi-object tracking results on representative frames.

4.5 Conclusion

In this chapter, we have addressed the multi-class multi-object model-free tracking task by formulate it as a graphical model inference problem, learning the parameters jointly and also find the most likely locations jointly. We also introduce a confidence parameter and motion model to overcome issues of sudden appearance change and occlusion. Comparing to several state-of-the-art models on challenging sequences, experimental results in term of single-object and multi-object evaluation protocols demonstrate effectiveness of the proposed method.

Chapter 5

Applying deep CNNs feature into multi-object model-free tracking

5.1 Introduction

In the previous chapter, we have modeled the multi-class multi-object model-free tracking as a PGM inference problem, in which only a hand-crafted HOG feature (Dalal and Triggs, 2005) has been applied to the method. While as shown in (Wang et al., 2015), feature representation in all aforementioned approaches in Chapter 2 also plays a crucial role in their performances. Conventionally, the hand-crafted features such as Haar-like features (Hare et al., 2011), HOG (Dalal and Triggs, 2005), subspace featues(Ross et al., 2008), color histograms(Zhao et al., 2010) have been used for this visual object tracking task. With the recent rise of deep learning and its superior performance in many visual recognition tasks (Girshick, 2015; Ren et al., 2015; He et al., 2017), the features extracted from CNNs has recently become popular and been extensively applied to visual tracking task, such as (Ma et al., 2015a; Henriques et al., 2015; Bertinetto et al., 2016). To this end, in this chapter, we also explore the importance of the feature representation on the performance of our proposed framework by comparing the popular hand-crafted HOG feature as well as CNNs based features.

In addition, we will conduct more extensively experiments to validate our proposed method, including extended datasets (from 12 to 24 sequences) and more experiments to investigate the functionality of different components of the proposed framework.

5.2 Overview of the proposed framework

In the multi-class multi-object model free tracking, given the bounding boxes of multiple objects of interest only in the first frame of the video, the tracker is expected to

track all targets from the 2nd frame to the end of the video.

We treat all targets of interest in each frame as a structured output modelled by an MRF, whose parameters are learned per frame while tracking the targets. Given the jointly learned parameters at each frame, the tracker localises all targets jointly via maximum a posteriori (MAP) inference maximising a score (global potential) consisting of both appearance and motion. The parameters are jointly learned in a maximum margin principle as in Struck (Hare et al., 2011), which essentially follows structured SVM (Tsochantaridis et al., 2005) and LaRank (Bordes et al., 2007).

5.2.1 Representation

The problem representation is following the definition (4.1) in 4.2.1, and still, we consider the objective function $f(\mathbf{x}^t, \mathbf{y}^t; \mathbf{w}^{(t-1)*})$ to be linear in some feature representation Φ, that is the same as (4.2).

5.2.2 Inference

The global feature $\Phi(\mathbf{x}^t, \mathbf{y}^t)$ for all targets in the *t-th* frames consists of local features involving only one or two targets as follows,

$$\Phi(\mathbf{x}^t, \mathbf{y}^t) = \begin{pmatrix} \Big(\phi_1(\mathbf{x}^t, y_i^t)\Big)_{i \in V} \\ \Big(\phi_2(\mathbf{x}^t, y_i^t)\Big)_{i \in V} \\ \Big(\phi_3(\mathbf{x}^t, y_i^t, y_j^t)\Big)_{(i,j) \in E} \end{pmatrix} \tag{5.1}$$

where $\phi_1(\mathbf{x}^t, \mathbf{y}_i^t)$ represents *i-th* target's appearance feature, $\phi_2(\mathbf{x}^t, \mathbf{y}_i^t)$ represents its motion feature, and $\phi_3(\mathbf{x}^t, y_i^t, y_j^t)$ represents the feature of edge $(i, j) \in E$.

Similarly, the parameters have 3 components

$$\mathbf{w}^{(t-1)*} = \begin{pmatrix} (\mathbf{u}_i^{(t-1)*})_{i \in V} \\ (\mathbf{p}_i^{(t-1)*})_{i \in V} \\ (\mathbf{v}_{i,j}^{(t-1)*})_{(i,j) \in E} \end{pmatrix} \tag{5.2}$$

where $(\mathbf{u}_i^{(t-1)*})_{i \in V}$, $(\mathbf{p}_i^{(t-1)*})_{i \in V}$ and $(\mathbf{v}_{i,j}^{(t-1)*})_{(i,j) \in E}$ are corresponding to the local features ϕ_1, ϕ_2 and ϕ_3, respectively.

Note both the second components of (5.1) and (5.2) are vectors, which are different from the corresponding components in (4.3) and (4.4).

Now objective function f, which is the same as (4.2), can be expressed as

$$f(\mathbf{x}^t, \mathbf{y}^t; \mathbf{w}^{(t-1)*}) = \sum_{i \in V} \underbrace{\Big(\langle \mathbf{u}_i^{(t-1)*}, \phi_1(\mathbf{x}^t, y_i^t) \rangle + \langle \mathbf{p}_i^{(t-1)*}, \phi_2(\mathbf{x}^t, y_i^t) \rangle \Big)}_{:=\theta_i(\mathbf{x}^t, y_i^t)} + \sum_{(i,j) \in E} \underbrace{\langle \mathbf{v}_{i,j}^{(t-1)*}, \phi_3(\mathbf{x}^t, y_i^t, y_j^t) \rangle}_{:=\theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t)} \tag{5.3}$$

Thus finding the most likely locations of all targets at each frame becomes a MRF inference problem as stated in (4.6)

5.2.3 Features and Potentials

Here we elaborate the features and potentials we used, in order to overcome challenges in model-free tracking, such as heavy occlusion and poor illumination.

5.2.3.1 Appearance and Motion as Node Potentials

Node potential function $\theta_i(\mathbf{x}^t, y_i^t)$ consists of two parts: the appearance model $F_{A,i}^t = \langle \mathbf{u}_i^{(t-1)*}, \phi_1(\mathbf{x}^t, y_i^t) \rangle$ and the motion model $F_{M,i}^t = \langle \mathbf{p}_i^{(t-1)*}, \phi_2(\mathbf{x}^t, y_i^t) \rangle$.

Appearance There are many choices for appearance feature $\phi_1(\mathbf{x}^t, y_i^t)$ as long as it represents the state appearance of *i-th* target. With such selected feature representation, the inner product $\langle \mathbf{u}_i^{(t-1)*}, \phi_1(\mathbf{x}^t, y_i^t) \rangle$ measures the similarity between the learned parameter $\mathbf{u}_i^{(t-1)*}$ and the state's appearance. This encourages the tracker to seek a location where the appearance in the current frame is close to that in the previous frame.

We separately incorporated two types of features in our tracker, HOG feature and the popular deep CNNs feature, to describe the state's appearance of targets, and our experiments will show the tracker's performance respectively.

HOG feature. HOG feature divides the image into relatively small cells and then calculate the histgram of oriented gradient in each local cell, resulting in its controllable degree of invariance to local geometric and photometric transformations (Dalal and Triggs, 2005). We apply this traditional HOG feature into our tracking algorithm to evaluate its performance.

Deep CNNs feature. To incorporate deep CNNs feature, we base our network on 19-layer *VGGNet* (*VGGNet-19*) (Simonyan and Zisserman, 2014), which is a pre-trained model on large-scale ImageNet (Russakovsky et al., 2015b), and also make

some modifications according to our proposed algorithm. The modifications will be elaborated in next section.

Architecture of the deep CNNs for feature extraction is as shown in Fig. 5.1.

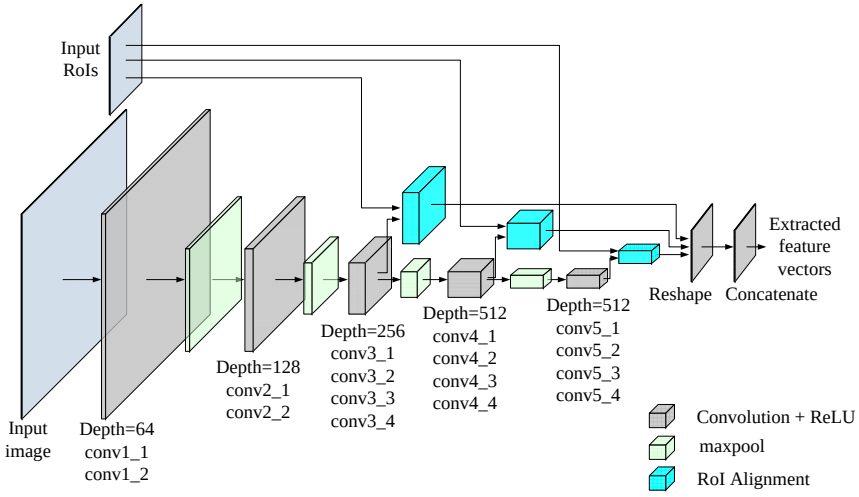

Figure 5.1: Network architecture. The architecture is based on *VGGNet-19*. The outputs from layer *conv3_4*, *conv4_4* and *conv5_4* are fed into RoI alignment layers, respectively, followed by reshape and concatenation layers to generate the final feature vectors

In the most left side of Fig. 5.1, instead of resize the input images, we feed the images at the original size into the *VGGNet-19*. As shown in (Ma et al., 2015a), along with forward propagation, the networks' strength in semantical discrimination between object categories is enhanced, while on the other hand, its spatial information of precise localization is weakened due to accumulated downsampling and pooling operations. So we follow the feature extraction trick of (Ma et al., 2015a), ignoring fully-connected layer and combining feature maps from 3 higher layers (*conv3_4*, *conv4_4* and *conv5_4*) for our tracking task. The sampled states in the input image are also mapped to these feature maps correspondingly. We treat each sampled state as a Region of Interest(RoI). In Fig. 5.1, the *RoI Alignment layer* (He et al., 2017) is an operation for extracting a small feature map for each RoI. It can properly align the extracted features with the input image, such characteristics makes this layer suitable to tracking tasks, which require pixel-accurate capacity. The feature map outputs from each *RoI Alignment layer* are then reshaped and concatenated to generate the feature vector.

Motion The motion model helps in two scenarios:

- When there is a sudden appearance change (such as heavy occlusion, sudden change of illumination or sudden pose change), the appearance model's value may dramatically drop for all candidates, and the tracker often loses tracking the targets relying on appearance alone. The motion model can continue to track all targets;
- When some candidates have similar appearance, the motion model helps to distinguish them via estimating their moving directions and positions, as shown in Fig. 5.2.

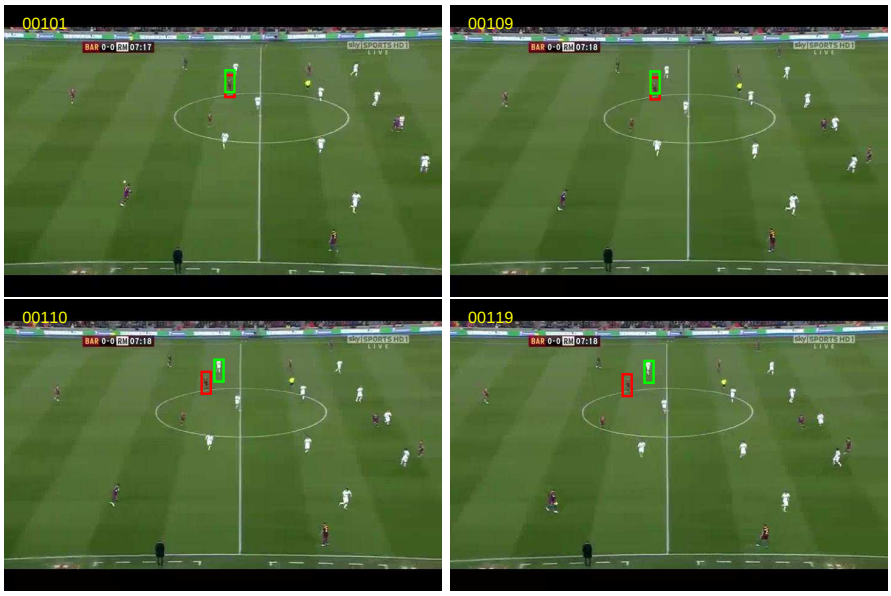

Figure 5.2: Comparison of tracking results, with and without motion model, for one of the players of interest. Red indicates the result with motion model, and green indicates the result without motion model

The motion feature $\phi_2(\mathbf{x}^t, y_i^t)$ is defined as a Gaussian kernel below

$$\begin{pmatrix} \exp\{-\alpha[(v_{i,x}^t - v_{i,x}^{(t-1)*})^2]\} \\ \exp\{-\alpha[(v_{i,y}^t - v_{i,y}^{(t-1)*})^2]\} \end{pmatrix} \tag{5.4}$$

to encourage the targets to be moving at a constant speed considering motion alone. Here $v_{i,x}^t$ and $v_{i,y}^t$ are estimated candidate's moving speed along X-axis and Y-axis, respectively. We simply use the offset at column and row coordinates, i.e. $v_{i,x}^{(t-1)*} = c_i^{(t-1)*} - c_i^{(t-2)*}$ and $v_{i,y}^{(t-1)*} = r_i^{(t-1)*} - r_i^{(t-2)*}$, by assuming that for two consecutive frames, the time gap is one unit, and the target's speed slightly varies only. The 2-d vector $\mathbf{p}_i^{(t-1)*}$ essentially becomes a trade-off parameter between the appearance model and the motion model.

5.2.3.2 Edge Potentials

Following 4.2.3.2, we use edge potentials $\theta_{i,j}(\mathbf{x}^t, y_i^t, y_j^t)$ to measure spatial relationship between two targets, and the specific definition follows (4.8)

5.2.4 Learning

The goal of learning is to find the optimal parameter $\mathbf{w}^{t*}$ in the current frame. Learning pricipally follows 4.2.4 to treat the learning as an optimization problem with restriction like (Tsochantaridis et al., 2005). By solving the dual problem (4.9), we can recover the optimal primal parameter via (4.11).

5.2.4.1 Discriminative Sampling

Here we explain at each frame, how we draw samples to form training data set. For each target, we draw five layers ellipses centered at the ground-truth location of the target. Note that this ground-truth location is given only at the first frame, and is predicted from the second frame onwards as all model free trackers do. The ellipses' major axis is in the direction of the estimated velocity. The ellipses define the state space for each target in a way slightly different from what we used in inference. We draw four types of samples as follows:

- the ground truth locations of all targets (only once);
- the ground truth location of one target, and randomly draw from the state spaces (on each of the ellipses) of the rest of the targets;
- the ground truth locations of two targets which are linked by an edge, and randomly draw from the state space (on each of the ellipses) of the rest of the targets;
- randomly draw from the state spaces (on each of the ellipses) of all targets.

Such sampling encourages the model to distinguish the true target from other targets and the background. This is particularly helpful when some targets look similar.

5.2.4.2 Confidence Parameter

When a sudden appearance change happens, we should not update the corresponding appearance parameter $\mathbf{u}_i^t$. We store the appearance similarity scores of each predicted target for the past m_k frames, a confidence parameter d_i^{t-1} is calculated as (4.13).

5.3 Datasets and evaluation metric

Compared to Chapter 4, we extend the test videos from 12 to 24, and more extensive experiments are conducted. In this section, we will first provide the information about the used datasets, then introduce the baselines, competitive methods, and the evaluation protocol.

Datasets. We evaluate the proposed multi-object tracking algorithm on 24 challenging sequences with varied object numbers (2-5 objects). Sequences 1-7 are obtained from dataset of SPOT (Zhang and van der Maaten, 2014), 8-10 are from dataset of MOT challenge (Ess et al., 2007; Leal-Taixé et al., 2015; Milan et al., 2016), 11-14 are from VOT 2016 (Kristan et al., 2016b) dataset[1], as for the rest, 15-24, are public sequences. The sequences, when needed, are manually annotated with the free video (image) annotation tool *ViTBAT* (Biresaw et al., 2016). These sequences include different challenging factors for multi-object tracking: very small targets, bad illumination, multiple interacting occlusions, appearance variations, complex deformations, etc. Summary of each sequence such as the number of targets and the number of frames is provided in Table 5.1.

The baselines and competitive methods. Similar to previous chapter 4.3, firstly, to demonstrate the functionality of different components of our tracker, we conduct four more baseline experiments including:

- without the edge potentials and confidence indicator, but with motion model (referred as *Ours WEWC*)
- without the edge potentials but with confidence indicator and motion model (referred as *Ours WE*)
- with confidence indicator and edge potentials, but without motion model (referred as *Ours WM*)
- without the confidence indicator, but with edge potentials and motion model (referred as *Ours WC*)

And to explore the contribution of deep CNNs feature, we also conduct experiments in which HOG feature and deep CNNs feature are incorporated, respectively.

To verify the performance of the proposed algorithm, we also compare it with six recent state-of-the-art trackers including: Siamese-FC (Bertinetto et al., 2016), CF (Ma et al., 2015a), KCF (Henriques et al., 2015), SPOT (Zhang and van der Maaten, 2014), Struck (Hare et al., 2011), ECO (Danelljan et al., 2017) and its variant ECO_HC.

[1]For sequences 8-14, we sliced some sequences to assure all objects of interest are occlusion-free in the first frame, and they are within the image throughout the whole sequence.

Table 5.1: Sequences summary

Sequence	Video name	Number of annotated objects	Challenges	Number of frames	Image size (Pixels)
1	airshow	4	similar appearance, occlusion, shaking camera	928	424 × 240
2	skydiving	5	similar appearance, panning camera	1237	656 × 480
3	flowers	4	appearance variation, multiple interacting occlusion	2249	360 × 240
4	hunting	2	dynamic moving severe appearance variation, occlusion	1755	480 × 266
5	shaking	3	bad illumination, appearance variation	344	624 × 352
6	parade	3	similar appearance, panning camera,	322	480 × 272
7	skating	2	bad illumination, complex deformation	150	640 × 360
8	MOT17-04	4	crowded, occlusion	458	960 × 540
9	MOT17-07	3	similar appearance	310	960 × 540
10	ETH_Crossing	2	occlusion	98	640 × 480
11	pedestrian2	3	very small targets, shaking camera	150	480 × 640
12	marching	2	similar appearance, occlusion	198	640 × 360
13	basketball	4	similar appearance, dynamic moving, complex deformations	334	576 × 432
14	birds2	3	similar appearance, occlusion	348	480 × 270
15	emu_run	5	dynamic moving, occlusion, shaking camera	186	640 × 480
16	Horse_racing	4	similar appearance, occlusion	160	320 × 240
17	man_kangaroo	2	different kind of targets, occlusion	315	640 × 480
18	Enter1cor	2	occlusion	275	384 × 288
19	soccer	5	very small targets, dynamic moving	620	720 × 480
20	Shop2cor	3	multiple interacting occlusion	250	384 × 288
21	F1	4	very small targets, appearance variation, occlusion	176	960 × 540
22	toddler_ducks	4	similar appearance, crowded	290	304 × 540
23	ducklings	5	similar appearance, multiple interacting occlusion	200	640 × 360
24	volleyball	3	similar appearance, dynamic moving, complex deformations	180	480 × 270

Siamese-FC uses fully convolutional Siamese network for tracking an object from background. CF is also a deep learning based approach, which incorporates the hierarchical CNN features into correlation filters, and tracks the object in a coarse-to-fine manner. KCF is one of the best correlation-filter-based trackers. Struck is a structural-learning-based tracker, which is closely related to our work and is in fact one of our baseline when we do not use motion model, joint learning and inference and our confidence parameter. ECO is developed based on the top ranked method of VOT2016, C-COT (Danelljan et al., 2016), and achieved 13% realative performance gain over C-COT, it incoporates fusion of deep CNNs feature and hand_crafted feature. ECO_HC is a fast variant of ECO, and it uses hand_crafted HOG (Dalal and Triggs, 2005) and ColorNames (van de Weijer et al., 2009b) features. Note that Siamese-FC, CF, KCF, Struck, ECO and ECO_HC are originally proposed for single object tracking, to track multiple objects with these algorithms, we simply extend them to track multi-object by running these trackers looping over the objects of inter-

est, meaning one loop corresponds to one target. SPOT is structural-learning-based multi-object tracking algorithm, we compare our algorithm to SPOT equipped with the minimum spanning tree.

Evaluation metrics. To assess the results, we employ two types of evaluation protocols: single-object tracking metrics and multi-object tracking metrics. For the former type, we use four metrics: 1) Center location error (CLE), which is computed as the average Euclidean distance between the ground-truth and the estimated center location of the tracked object. 2) VOC overlap ratio (VOR), which is defined as $R(B_T \cap B_{GT})/R(B_T \cup B_{GT})$, where $R(B)$ measure the area of the bounding box B, B_T and B_{GT} are the tracking bounding box and the ground truth bounding box, respectively. 3) Precision plot, which measures the percentage of successfully tracked frames. Tracking on a frame is considered successful if the distance between the centers of the predicted box and the ground truth box is under a fixed threshold. 4) Success plot, which is defined as the percentage of frames where VOC overlap ratio is larger than a certain threshold (Wu et al., 2013). For multi-object tracking metrics, we follow the evaluation protocol of (Milan et al., 2016), where five of those metrics are used: Recall represents the percentage of the detected targets, IDS represents mismatch error, FAR represents the number of false alarms per frame, MT represents the number of mostly tracked targets, MOTA means multi-object tracking accuracy, and MOTP means multi-object tracking precision.

5.4 Experiments

In this section, we will elaborate implementation details of our experiments, and then present quantitative evaluation, as well as qualitative evaluation, of the proposed method.

5.4.1 Implementation Details

Similar to visual tracker (Hare et al., 2011), the proposed tracker performs object localization using a sliding-window search scheme with a adaptive search radius. Generally, for each target, we search in a 2D area with $\{(\Delta c_i, \Delta r_i)|(\Delta c_i)^2 + (\Delta r_i)^2 \leq R_i^2\}$, and sample on a polar grid. We use 5 radial and 16 angular divisions, giving 81 locations, which are used for discriminative sampling. In our implementation, the $2nd$ to $4th$ types of samples in discriminative sampling are repeated 27 times, 16 times and 81 times, respectively. As different targets may be different sizes, to acquire effective samples, we set $R_i = 2 \times min(w_i^t, h_i^t)$. When the motion model applies, we sample within an ellipse whose major axis aligns with the direction of

the target's velocity, and its length depends R_i and target's speed. As for confidence parameter, we set up the threshold $d_i^{t-1} = 0.1$ based on the past $m_k = 5$ frames, accordingly the score coefficients μ_j are set to 0.4, 0.3, 0.2 and 0.1, respectively. Also we fix hyper-parameters $\alpha = 0.05$ for motion feature and $\lambda = 0.025$ and $\gamma = 0.015$ for edge potentials , respectively.

Feature representation is so crucial that both deep CNNs feature and histogram of oriented gradients (HOG) (Dalal and Triggs, 2005) feature are incorporated in our tracking algorithm, and we will evaluate their performance separately.

When deep CNNs feature is applied, RoI alignment operation is neccessary. Following the spatial pyramid pooling (He et al., 2014), we set different level pooling bins for convolutional layers *conv3_4*, *conv4_4*, and *conv5_4*, Table 5.2 states the details of pooling configuration, and Fig. 5.3 takes feature map *conv3_4* for example to elaborate the pooling operation.

The last implementation detail is graph structure. we use fully connected graphs in images where targets are expected to not move very fast (i.e. since maintaining distances and angles between targets in that case is better), and use simple chain structure otherwise.

Table 5.2: The table shows RoI alignment pooling configuration for deep CNNs feature extraction

Layers	pooing level	pooling bin(s)
conv3_4	2	4×4, 1×1
conv4_4	2	2×2, 1×1
conv5_4	1	1×1

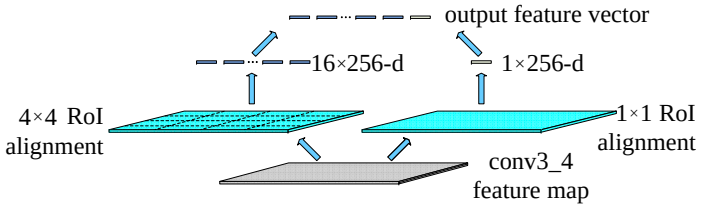

Figure 5.3: Illustration of detailed operation to generate feature vector of layer *conv3_4*

To perform a fair comparison with peer trackers SPOT and KCF, we also extract the same features, i.e. Histogram of oriented gradients (HOG)(Dalal and Triggs, 2005), for tracking multiple objects. For our tracker, we use fully connected graphs in images where targets are expected to not move very fast (i.e. since maintaining distances and angles between targets in that case is better), and use simple chain

structure otherwise.

5.4.2 The state space for MRF inference

As aforementioned, jointly predicting the most likely locations of all targets can be cast as a MRF inference problem. Generally, definition of the state space for the inference follows 4.4.2,i.e. when the motion model is used, the state space for the MRF inference is defined by a circle with radius $\bar{R}$, where $\bar{R}$ (pixels) is a prefixed constant; while when the motion model is used, we take the target location at the previous frame as the center, and draw an ellipse whose major axis follows the direction of the estimated velocity of the target. The state space is defined by an ellipse with its parameters such as semi major, semi minor and its rotation angle as (4.14) and (4.17).

An illustration of the ellipse for the i-th target is shown in Fig.4.2 depending on the velocity of the target. Same as the case without the motion model, every second pixel within the ellipse is considered as a feasible state.

5.4.3 Sampling training data

Sampling training data for parameter learning is following the method stated in 4.4.3, i.e. when the motion model is disabled, for the i-th target at the t-th frame, the sampling space for structured SVM learning is empirically defined by 5 concentric circles with the most outer layer radius defined as

$$R_i^t = 1.2 \times \min(w_i^t, h_i^t)$$

to accommodate the fact that different targets often have different sizes in multi-object tracking scenario; while when the motion model is enabled, we use 5 ellipses shown in Fig. 4.3 to define the candidates of the training data to be sampled.

All 5 ellipses share one center and the same rotation angle. Semi major and semi minor of the most outer ellipse (layer 1) are calculated as (4.18),

5.4.4 Quantitative Evaluation

We first carry out the experiments to show the contribution of each components in our algorithm and then compare the proposed method with the aforementioned state-of-the-art methods.

Evaluation on baselines of functional components. To explore the effectiveness of different components of our tracker, we report the results of proposed algorithm with each component switched on or off in the tables. The results in Tables 5.3 and 5.4 show that each component consistently contribute in improving the total

Table 5.3: Quantitative evaluation of the proposed algorithm with 4 baselines on the reported sequences. We report the average VOR and CLE of each tracker for all objects in each sequences, and their average by all sequences (the last row in the table). The best results are shown in bold.

Sequence	*Ours WEWC*		*Ours WE*		*Ours WM*		*Ours WC*		*Ours HOG*		*Ours CNNs*	
	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE
airshow	0.74	6.5	0.73	6.6	0.74	6.3	0.74	6.5	0.74	6.5	**0.76**	**5.6**
skydiving	0.74	16.0	0.71	14.8	0.74	8.3	**0.81**	**4.3**	0.81	4.6	0.80	4.7
flowers	0.53	30.4	0.51	33.6	0.70	14.5	0.70	12.9	0.72	11.8	**0.79**	**7.9**
hunting	0.37	127.2	0.42	112.7	0.17	174.9	0.50	44.8	0.71	16.3	**0.74**	**13.6**
parade	0.79	4.2	0.79	4.3	0.81	4.0	0.79	4.1	0.79	4.3	**0.83**	**3.3**
shaking	0.69	26.8	0.78	9.5	0.78	11.7	0.78	11.5	0.83	**5.9**	**0.83**	6.1
skating	0.57	21.9	0.60	20.2	0.61	21.3	0.64	19.5	0.68	13.6	**0.78**	**9.5**
MOT17-04	0.77	38.2	0.77	38.2	0.87	3.2	0.87	3.0	**0.87**	**3.0**	0.86	3.3
MOT17-07	**0.86**	**4.9**	0.86	5.1	0.86	5.0	0.86	5.0	0.86	5.1	0.86	5.3
ETH_Crossing	0.58	92.5	0.80	**13.8**	0.78	16.0	0.77	16.1	0.76	18.5	**0.81**	15.9
pedestrian2	0.83	1.8	0.83	1.8	0.85	**1.5**	**0.85**	1.5	0.80	2.0	0.84	1.6
marching	0.81	6.6	0.81	6.6	0.78	10.6	0.81	6.8	0.81	6.8	**0.82**	**6.4**
basketball	0.73	11.1	0.74	9.5	0.75	13.7	0.74	10.7	0.75	8.5	**0.82**	**7.4**
birds2	0.64	9.4	0.65	9.0	0.63	9.6	0.67	7.6	0.69	6.7	**0.76**	**5.3**
emu_run	0.80	4.4	0.80	4.4	**0.83**	**3.0**	0.82	3.1	0.80	4.4	0.78	5.0
Horse_racing	0.68	4.6	0.68	4.7	0.70	4.0	0.71	3.7	0.71	3.7	**0.73**	**3.6**
man_kangaroo	0.67	28.7	0.79	13.1	0.69	31.4	0.80	12.2	0.80	**11.6**	**0.80**	13.7
Enter1cor	0.85	5.1	0.86	4.6	0.86	4.9	0.86	4.7	0.86	4.7	**0.89**	**3.5**
soccer	0.65	17.6	0.65	13.5	0.63	21.0	0.75	3.8	0.76	3.0	**0.79**	**2.4**
Shop2cor	0.81	7.9	0.81	7.9	**0.81**	7.8	0.81	**7.5**	0.81	7.9	0.80	8.7
F1	0.77	8.2	0.77	8.2	0.56	35.9	**0.81**	**1.7**	0.78	2.5	0.80	2.0
toddler_ducks	0.74	6.3	0.74	6.0	0.75	4.4	**0.78**	4.1	0.78	**4.1**	0.77	4.5
ducklings	0.67	10.9	0.67	10.9	0.75	3.8	0.75	3.6	**0.78**	**3.1**	0.72	4.1
volleyball	0.38	44.1	0.36	38.3	0.41	16.1	0.48	16.1	0.57	7.6	**0.80**	**3.2**
Average	0.69	22.3	0.71	16.5	0.71	18.0	0.75	9.0	0.77	6.9	**0.80**	**6.1**

Table 5.4: The table shows baseline multi-object tracking results of baseline methods in terms of MOT performance criteria on all sequences.

Tracker	Rcll	FAR	MT	IDS	MOTA	MOTP
Ours WEWC	80.5	0.69	57	27	60.6	79.2
Ours WE	80.7	0.68	60	18	61.2	79.0
Ours WM	83.8	0.58	63	21	67.4	79.4
Ours WC	90.9	0.32	67	11	81.6	79.0
Ours HoG	94.1	0.21	75	13	88.0	78.9
Ours CNNs	**97.0**	**0.11**	**76**	**5**	**94.0**	**80.4**

performance of the proposed method with respect to all reported metrics. For example, considering the initial baseline as "*Ours WEWC*", we can see from Table 5.4 that incorporating confidence parameter only ("*Ours WE*"), edge potential term (joint learning and inference) only ("*Ours WC*") and both terms together ("*Ours HOG*") can respectively improve the results with respect to MOTA metric by 0.9%, 34.7% and 45.2%, and the MOTA metric will be improved by 55.1% if deep CNNs feature is applied ("*Ours CNNs*").

Note in Chapter 4, our motion feature $\phi_2(\mathbf{x}^t, y_i^t)$ is defined as a scalar Gaussian kernel in (4.7), and in this chapter, we improve and define it as a vector in (5.4), which is based on the speeds along X-axis and Y-axis, respectively. We also conduct experiments under the same conditions (i.e. same sequences, same objects of interest, etc) as stated in Chapter 4, compared to the old version motion feature (4.7), *Ours HOG* achieved improvement by 3.8% (0.81 vs. 0.78) on VOR and 1.3% (7.1 vs. 7.2) on CLE, repectively; and correspondingly, the MOT metrics are also improved: e.g. Recall increased by 1.8% (96.0 vs. 94.4) and MOTA increased by 5.6% (93.6 vs. 88.6). This fact shows that our improved motion model works as expected.

Comparison with state-of-the-art methods. Tables 5.5 and 5.6 show comparison between our proposed approach and the aforementioned competitive methods on the reported sequences using average CLE and VOR, and also MOT criteria. The proposed method significantly improves the baseline structural-learning-based tracker. Our tracker, with deep CNN features, obtains almost the best performance on most sequences among all the compared trackers with an average CLE of 6.1 pixels and average VOR with 0.80.

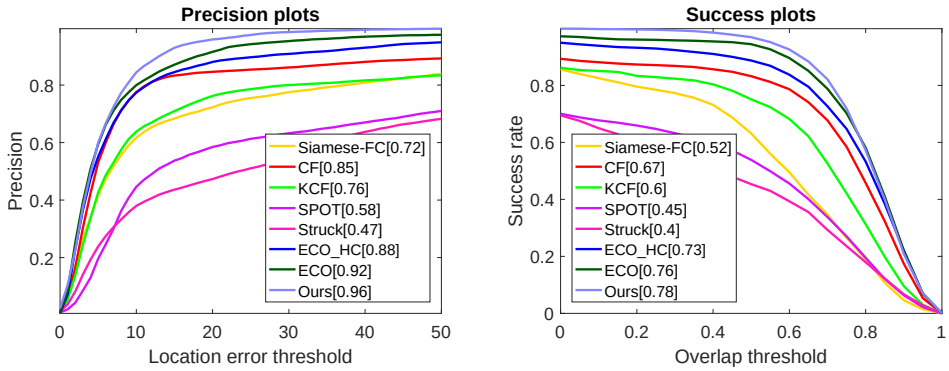

Figure 5.4: Overall distance precision plot (left) and overlap success plot (right) over all sequences. The legends show the precision scores and AUC scores for each tracker.

As shown in Fig. 5.4, the precision plot and success plot for all dataset indicate that our method outperforms five other state-of-the-art trackers. All algorithms are

compared in terms of the same initial positions in the first frame. For precision plots and success plots, the values in the legend are the average distance precision at 20 pixels and the Area Under Curve (AUC) scores, respectively. In both plots, the best method is the proposed tracker; our algorithm ("*Ours CNNs*") outperforms the competitors in both mean distance precision and mean AUC scores, respectively.

In Table 5.6, we compare the performance of our proposed tracker with the competitive trackers using multi-object tracking (MOT) metrics. Table 5.6 shows that our method outperforms other methods overall. The MOTA is perhaps the most widely used metric to evaluate a multi-object tracker's performance (Bernardin and Stiefelhagen, 2008b). It takes three sources of errors (i.e. the number of false positive, of misses, and of mismatches). Note that a negative MOTA means that the number of errors made by the tracker exceeds the number of all objects. The MOTP is the average dissimilarity between all true positives and their corresponding ground truth targets. The higher the Recall, MT, MOTA and MOTP of a tracker, the better its performance. On the contrary, the lower FAR and IDs indicates a better performance. In Table 5.6, the state-of-the-art ECO (Danelljan et al., 2017), which fuses deep CNNs feature and hand_crafted feature, outperforms the other existing methods in our evaluation, it reveals the excellent design of ECO and the power of feature fusion. By incorporating effective motion model and spatial information, our tracker,with HOG feature (*Ours HOG*) and deep CNNs feature (*Ours CNNs*), outperforms ECO_HC and ECO by 4.1% and 0.4% in MOTA, respectively. Moreover, the application of deep CNNs feature in our tracker (*Ours CNNs*) yields a 6.8% overall improvement over the tracker with traditional HOG feature (*Ours HOG*).

5.4.5 Qualitative Evaluation

We present several selected tracking results from the proposed tracker in different rows of Fig. 5.5. For presentation clarity, the tracking results of the other trackers are not shown in the figure. Overall, our tracker is able to predict precise localization for the multiple objects of interest. Note that sequence *skydiving* (the 1st row) and sequence *MOT17-04* (the 5th row) show the proposed multi-object tracker keeps tracking the objects of interest when the background is cluttered and crowded, because it leverages the structural information encoded in the graph edge potentials; sequence *flowers* (the 2nd row), sequence *basketball* (the 6th row) and sequence *volleyball* (the last row) validates that the proposed tracker works well under circumstances of deformations, appearance variations and multiple interacting occlusion; sequence *shaking* (the 3rd row) and sequence *shaking* (the 4th row) verify the performance of the proposed tracker in presence of bad illumination; sequence *soccer* (the 8th

Table 5.5: Quantitative evaluation of the proposed algorithm with 6 state-of-the-art methods on the reported sequences. We report the average VOR and CLE of each tracker for all objects in each sequences, and their average by all sequences (the last row in the table). The best results are shown in bold.

Sequence	*Siamese-FC*		*CF*		*KCF*		*SPOT*		*Struck*		*ECO_HC*		*ECO*		*Ours*	
	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE	VOR	CLE
airshow	0.60	7.9	0.74	6.5	**0.81**	**4.3**	0.70	7.5	0.49	17.1	0.79	4.8	0.80	4.7	0.76	5.6
skydiving	0.40	28.1	0.75	6.1	0.75	5.8	0.71	7.7	0.36	30.8	0.80	4.6	**0.82**	**4.2**	0.80	4.7
flowers	0.41	24.7	0.73	12.3	0.64	14.9	0.67	15.9	0.15	86.4	**0.84**	**5.5**	0.83	6.2	0.79	7.9
hunting	0.19	94.9	0.07	131.4	0.01	228.5	0.07	127.6	0.19	83.6	0.58	26.2	0.73	15.0	**0.74**	**13.6**
parade	0.51	14.3	0.81	3.6	0.79	4.1	0.54	9.7	0.52	28.8	**0.83**	**2.9**	0.79	3.4	0.83	3.3
shaking	0.75	7.2	0.60	44.8	0.54	31.6	0.67	14.8	0.43	32.4	0.77	8.6	0.78	8.8	**0.83**	**6.1**
skating	0.37	83.4	**0.85**	**6.2**	0.52	81.4	0.53	42.5	0.44	101.4	0.64	26.5	0.74	11.1	0.78	9.5
MOT17-04	0.24	120.2	0.71	40.5	0.63	53.7	0.59	43.1	0.35	82.0	0.85	3.9	0.75	50.7	**0.86**	**3.3**
MOT17-07	0.70	10.6	0.86	5.2	0.74	12.1	0.83	5.1	0.78	9.2	0.89	**4.6**	**0.89**	4.7	0.86	5.3
ETH_Crossing	0.62	22.3	0.63	52.8	0.64	34.8	0.65	30.3	0.47	74.9	**0.81**	**13.9**	0.81	14.1	0.81	15.9
pedestrian2	0.66	6.2	0.60	15.3	0.56	18.7	0.01	69.5	0.41	24.6	0.82	**1.6**	0.83	1.7	**0.84**	1.6
marching	0.49	26.5	0.70	13.5	0.79	7.4	0.57	17.4	0.13	102.1	0.73	15.3	0.73	13.7	**0.82**	**6.4**
basketball	0.36	94.7	**0.85**	**5.1**	0.47	59.6	0.15	128.4	0.26	113.3	0.65	39.3	0.80	8.9	0.82	7.4
birds2	0.50	18.1	0.49	15.9	0.36	24.0	0.62	8.2	0.34	41.6	0.53	20.8	0.54	20.6	**0.76**	**5.3**
emu_run	0.63	5.6	0.76	5.3	0.74	4.7	0.52	23.1	0.54	12.3	0.84	2.9	**0.85**	**2.5**	0.78	5.0
Horse_racing	0.65	4.4	0.70	3.8	0.69	**3.1**	0.58	7.1	0.37	23.6	0.71	3.5	0.70	3.6	**0.73**	3.6
man_kangaroo	0.51	28.9	0.34	139.0	0.29	79.5	0.37	65.1	0.49	47.5	0.75	15.3	0.73	14.8	**0.80**	**13.7**
Enter1cor	0.60	5.9	0.85	4.9	0.84	5.9	0.82	5.5	0.64	8.3	0.89	3.8	**0.89**	**3.4**	0.89	3.5
soccer	0.72	2.8	0.72	2.9	0.64	21.4	0.09	121.1	0.57	17.2	0.78	2.6	0.77	2.7	**0.79**	**2.4**
Shop2cor	0.47	15.3	0.62	27.3	0.47	32.6	0.57	32.0	0.55	19.5	0.71	11.6	0.70	12.1	**0.80**	**8.7**
F1	0.58	2.3	0.73	2.9	0.80	2.0	0.01	278.3	0.13	214.1	0.64	63.5	**0.85**	**1.3**	0.80	2.0
toddler_ducks	0.69	5.0	0.80	**3.4**	0.55	16.8	0.25	53.2	0.47	26.4	0.75	12.1	**0.83**	3.7	0.77	4.5
ducklings	0.64	8.7	0.76	3.5	**0.79**	**2.8**	0.08	96.9	0.36	22.6	0.62	8.2	0.78	3.1	0.72	4.1
volleyball	0.43	54.2	0.78	3.5	0.67	9.0	0.27	26.1	0.35	50.0	0.59	20.1	0.67	11.8	**0.80**	**3.2**
Average	0.53	28.8	0.69	23.2	0.61	31.6	0.45	51.5	0.41	52.9	0.74	13.4	0.78	9.4	**0.80**	**6.1**

Table 5.6: The table shows multi-object tracking results of competitive methods in terms of MOT performance criteria on all sequences.

Tracker	Recall	FAR	MT	IDs	MOTA	MOTP
Siamese-FC	57.0	1.53	38	33	13.5	71.5
CF	88.7	0.40	67	3	77.4	78.8
KCF	77.8	0.79	58	18	55.4	75.2
SPOT	67.3	1.16	34	24	34.3	74.3
Struck	43.9	1.99	17	110	-13.5	75.0
ECO_HC	92.3	0.27	70	**2**	84.5	**81.3**
ECO	96.8	0.11	**76**	3	93.6	81.2
Ours HOG	94.1	0.21	75	13	88.0	78.9
Ours CNNs	**97.0**	**0.11**	76	5	**94.0**	80.4

Figure 5.5: Multi-object tracking results on representative sequences. From top to bottom: skydiving, flowers, shaking, skating, basketball, man_kangaroo, soccer, F1 and volleyball.

row) and sequence *F1* (the 9th row) show our tracker's performance in tracking very small targets; while sequence *man_kangaroo* (the 7th row) demonstrates our tracker's capability in tracking different type of targets simultaneously. The range of these sequences and different challenges presented in each, demonstrate that our tracker performs well in various circumstances.

5.5 Conclusion

In this chapter, we have addressed the multi-class multi-object model-free tracking task by formulate it as a graphical model inference problem, and find the most likely locations jointly. We also explore the capability of popular deep CNNs features and develop a dataset for this task . Comparing to several competitive models, including the state-of-the-art, on challenging sequences, experimental results in term of single-object and multi-object evaluation protocols demonstrate effectiveness of the proposed method.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we have proposed joint learning and joint inference for multiple generic object tracking, and also generated a dataset for such task. To this end, Chapter 4 has introduced an approach for a joint learning and joint inference framework, together with a motion model, for tracking multiple generic objects, and also built a basic dataset for this task, which includes 12 videos; Chapter 5 has improved aforementioned model, applied popular deep CNNs feature into the model and extended the dataset to 24 videos.

6.2 Future work

Beyond the solved problem elaborated in previous chapters, there are several important issues remaining for the multi-class multi-object tracking task. Here we discuss some potential future directions.

- Initiation and termination

  In the models and solutions stated in Chapter 4 and Chapter 5, there is an assumption that all the objects of interest stay in the view throughout the whole video. Such assumption, to some degree, restricts the application of the proposed method. We will design some mechanism to eliminate such restriction, that is, the proposed method can handle the case that targets moving in and out of the view. This will lead to much wider application of the proposed method.

- Non-linear objective function

  There is another assumption, for the models, that the objective function is linear with regard to the input feature representation. So it is possible to extend this linear objective function to non-linear case, (*e.g.*, incorporating deep neural networks in objective function.

- Expansion of dataset

  Although the dataset has been extended to 24 videos, it is still small, especially compared to popular SOT dataset. We are interested in expanding the dataset with more challenging videos, especially those can highlight multi-class multi-object tracking attributes. We will make this dataset public.

Bibliography

Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; and Savarese, S., 2016. Social LSTM: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 15)

Austvoll, I. and Kwolek, B., 2010. Region covariance matrix-based object tracking with occlusions handling. In *Proceedings of the 2010 International Conference on Computer Vision and Graphics: Part I*, ICCVG'10 (Warsaw, Poland, 2010), 201–208. Springer-Verlag, Berlin, Heidelberg. (cited on page 7)

Babenko, B.; Yang, M.-H.; and Belongie, S., 2011. Robust object tracking with online multiple instance learning. *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 33, 8 (2011), 1619–1632. (cited on pages 1, 6, 7, 8, and 31)

Bakir, G. H.; Hofmann, T.; Schölkopf, B.; Smola, A. J.; Taskar, B.; and Vishwanathan, S. V. N., 2007. *Predicting Structured Data (Neural Information Processing)*. The MIT Press. ISBN 0262026171. (cited on page 20)

Berclaz, J.; Fleuret, F.; Turetken, E.; and Fua, P., 2011. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 9 (Sept 2011), 1806–1819. doi:10.1109/TPAMI.2011.21. (cited on pages 1 and 14)

Bernardin, K. and Stiefelhagen, R., 2008a. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 1 (May 2008). doi:10.1155/2008/246309. https://doi.org/10.1155/2008/246309. (cited on page 16)

Bernardin, K. and Stiefelhagen, R., 2008b. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 1 (2008), 1–10. (cited on pages 46 and 64)

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H., 2016. Fully-convolutional siamese networks for object tracking. In *2016 European Conference on Computer Vision (ECCV)*, 850–865. Springer. (cited on pages 6, 29, 39, 45, 51, and 57)

Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36, 2 (1974), pp. 192–236. http://www.jstor.org/stable/2984812. (cited on page 23)

Biresaw, T. A.; Nawaz, T.; Ferryman, J.; and Dell, A., 2016. Vitbat: Video tracking and behavior annotation tool. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, (2016). (cited on pages 38 and 57)

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc. ISBN 0387310738. (cited on pages 21 and 24)

Blaschko, M. B. and Lampert, C. H., 2008. Learning to localize objects with structured output regression. *2008 European Conference on Computer Vision (ECCV)*, 5302 (2008), 2–15. (cited on page 37)

Bordes, A.; Bottou, L.; Gallinari, P.; and Weston, J., 2007. Solving multiclass support vector machines with larank. *Proceedings of the 24th International Conference on Machine learning*, (2007). (cited on pages 2, 20, 32, 33, 36, 37, and 52)

Borji, A.; Frintrop, S.; Sihite, D. N.; and Itti, L., 2012. Adaptive object tracking by learning background context. In *2012 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 23–30. doi:10.1109/CVPRW.2012.6239191. (cited on page 9)

Breitenstein, M. D.; Reichlin, F.; Leibe, B.; Koller-Meier, E.; and Gool, L. V., 2009. Robust tracking-by-detection using a detector confidence particle filter. In *2009 IEEE International Conference on Computer Vision (ICCV)*, 1515–1522. doi:10.1109/ICCV.2009.5459278. (cited on page 14)

Brendel, W.; Amer, M.; and Todorovic, S., 2011. Multiobject tracking as maximum weight independent set. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1273–1280. doi:10.1109/CVPR.2011.5995395. (cited on page 15)

Butt, A. A. and Collins, R. T., 2013. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1846–1853. doi:10.1109/CVPR.2013.241. (cited on page 1)

Chen, D. S. and Liu, Z. K., 2007. Generalized haar-like features for fast face detection. In *2007 International Conference on Machine Learning and Cybernetics*, vol. 4, 2131–2135. doi:10.1109/ICMLC.2007.4370496. (cited on page 8)

Chen, K.; Gong, S.; and Xiang, T., 2011. Human pose estimation using structural support vector machines. In *2011 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 846–851. (cited on page 21)

Chen, X.; Treiber, M.; Kanagaraj, V.; and Li, H., 2018. Social force models for pedestrian traffic - state of the art. *Transport Reviews*, 38, 5 (2018), 625–653. doi: 10.1080/01441647.2017.1396265. (cited on page 15)

Choi, W. and Savarese, S., 2012. A unified framework for multi-target tracking and collective activity recognition. In *2012 European Conference on Computer Vision (ECCV)*. (cited on pages 13 and 14)

Chu, Q.; Ouyang, W.; Li, H.; Wang, X.; Liu, B.; and Yu, N., 2017. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 4846–4855. (cited on pages 8 and 29)

Clifford, P., 1990. Markov random fields in statistics. In *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley* (Eds. G. Grimmett and D. Welsh), 19–32. Oxford University Press, Oxford. (cited on page 23)

Comaniciu, D.; Ramesh, V.; and Meer, P., 2003. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 5 (May 2003), 564–575. doi:10.1109/TPAMI.2003.1195991. https://doi.org/10.1109/TPAMI.2003.1195991. (cited on page 5)

Crammer, K. and Singer, Y., 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2 (Mar. 2002), 265–292. http://dl.acm.org/citation.cfm?id=944790.944813. (cited on page 19)

Dai, Y. and Liu, B., 2015. Robust video object tracking using particle filter with likelihood based feature fusion and adaptive template updating. *arXiv preprint arXiv:1509.08182, 2013. URL: arXiv:1509.08182*, (2015). (cited on page 9)

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2005), 886–893. (cited on pages 1, 5, 8, 32, 40, 51, 53, 58, and 60)

Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; and Felsberg, M., 2017. Eco: Efficient convolution operators for tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 6, 8, 9, 57, and 64)

Danelljan, M.; Robinson, A.; Khan, F. S.; and Felsberg, M., 2016. Beyond correlation filters: Learning continuous convolution operatorsfor visual tracking. In *European Conference on Computer Vision (ECCV)*, 472–488. doi:10.1007/978-3-319-46454-1_29. https://doi.org/10.1007/978-3-319-46454-1_29. (cited on pages 6, 8, and 58)

Dinh, T. B.; Vo, N.; and Medioni, G., 2011. Context tracker: Exploring supporters and distracters in unconstrained environments. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 00, 1177–1184. doi.ieeecomputersociety.org/10.1109/CVPR.2011.5995733. (cited on page 9)

Dollár, P.; Wojek, C.; Schiele, B.; and Perona, P., 2009. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 16)

Dollár, P.; Wojek, C.; Schiele, B.; and Perona, P., 2012. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (2012). (cited on page 16)

Ess, A.; Leibe, B.; and Gool, L., 2007. Depth and appearance for mobile scene analysis. In *2007 IEEE International Conference on Computer Vision (ICCV)*. (cited on page 57)

Fablet, R. and Black, M. J., 2002. Automatic detection and tracking of human motion with a view-based representation. In *2002 European Conference on Computer Vision (ECCV)*, vol. 1 of *LNCS 2353*, 476–491. Springer-Verlag. (cited on page 7)

Fan, J.; Wu, Y.; and Dai, S., 2010. Discriminative spatial attention for robust tracking. In *2010 European Conference on Computer Vision (ECCV)*, 480–493. Springer Berlin Heidelberg, Berlin, Heidelberg. (cited on page 8)

Faux, F. and Luthon, F., 2012. Theory of evidence for face detection and tracking. *International Journal of Approximate Reasoning*, 53, 5 (2012), 728–746. (cited on page 7)

Fortmann, T. E.; Bar-Shalom, Y.; and Scheffe, M., 1980. Multi-target tracking using joint probabilistic data association. In *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 807–812. doi:10.1109/CDC.1980.271915. (cited on page 1)

Geiger, A.; Lenz, P.; and Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 16)

Ghasemi, A. and Safabakhsh, R., 2012. A real-time multiple vehicle classification and tracking system with occlusion handling. In *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*, 109–115. doi:10.1109/ICCP.2012.6356172. (cited on page 13)

Girshick, R., 2015. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 29 and 51)

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 28)

Grabner, H.; Grabner, M.; and Bischof, H., 2006. Real-time tracking via on-line boosting. In *British Machine Vision Conference (BMVC)*, 47–56. British Machine Vision Association. (cited on pages 6, 8, and 9)

Grabner, H.; Matas, J.; Gool, L. J. V.; and Cattin, P. C., 2010. Tracking the invisible: Learning where the object might be. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1285–1292. doi:10.1109/CVPR.2010.5539819. https://doi.org/10.1109/CVPR.2010.5539819. (cited on page 9)

Gu, S.; Zheng, Y.; and Tomasi, C., 2011. Linear time offline tracking and lower envelope algorithms. In *2011 IEEE International Conference on Computer Vision (ICCV)*, ICCV '11, 1840–1846. IEEE Computer Society. doi:10.1109/ICCV.2011.6126451. http://dx.doi.org/10.1109/ICCV.2011.6126451. (cited on page 6)

Hare, S.; Saffari, A.; and Torr, P. H., 2011. Struck: Structured output tracking with kernels. *2011 IEEE International Conference on Computer Vision (ICCV)*, (2011). (cited on pages 2, 5, 6, 7, 8, 9, 21, 32, 34, 36, 39, 40, 45, 51, 52, 57, and 59)

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B., 2017. Mask R-CNN. *CoRR*, abs/1703.06870 (2017). http://arxiv.org/abs/1703.06870. (cited on pages 29, 51, and 54)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729 (2014). http://arxiv.org/abs/1406.4729. (cited on page 60)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. 770–778. (cited on page 28)

Helbing, D. and Molnár, P., 1995. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51 (May 1995), 4282–4286. doi:10.1103/PhysRevE.51.4282. https://link.aps.org/doi/10.1103/PhysRevE.51.4282. (cited on page 15)

Held, D.; Thrun, S.; and Savarese, S., 2016. Learning to track at 100 fps with deep regression networks. In *2016 European Conference Computer Vision (ECCV)*. (cited on page 8)

Henriques, J. F.; Caseiro, R.; Martins, P.; and Batista, J., 2015. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 3 (2015), 583–596. (cited on pages 39, 45, 51, and 57)

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580 (2012). (cited on page 28)

Hong, S. and Han, B., 2014. Visual tracking by sampling tree-structured graphical models. In *2014 European Conference on Computer Vision (ECCV)*, 1–16. (cited on page 24)

Hu, M.; Ali, S.; and Shah, M., 2008. Detecting global motion patterns in complex videos. *2008 19th International Conference on Pattern Recognition*, (2008), 1–5. (cited on page 15)

Hu, W.; Li, X.; Luo, W.; Zhang, X.; Maybank, S.; and Zhang, Z., 2012. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 12 (Dec 2012), 2420–2440. doi:10.1109/TPAMI.2012.42. (cited on pages 13 and 16)

Hua, Y.; Alahari, K.; and Schmid, C., 2014. Occlusion and motion reasoning for long-term tracking. In *2014 European Conference on Computer Vision (ECCV)*. (cited on page 6)

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15 (Lille, France, 2015), 448–456. JMLR.org. (cited on page 28)

Iqbal, U.; Milan, A.; and Gall, J., 2017. PoseTrack: Joint multi-person pose estimation and tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 13)

Izadinia, H.; Saleemi, I.; Li, W.; and Shah, M., 2012. (mp)2t: Multiple people multiple parts tracker. In *2012 European Conference on Computer Vision (ECCV)* (Florence, Italy, 2012), 100–114. Springer-Verlag, Berlin, Heidelberg. doi:10.1007/

978-3-642-33783-3_8. http://dx.doi.org/10.1007/978-3-642-33783-3_8. (cited on page 14)

Jia, X.; Lu, H.; and Yang, M. H., 2012. Visual tracking via adaptive structural local sparse appearance model. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1822–1829. doi:10.1109/CVPR.2012.6247880. (cited on page 9)

Jordan, M. I. (Ed.), 1999. *Learning in Graphical Models*. MIT Press. ISBN 0-262-60032-3. (cited on page 21)

Jordan, M. I., 2004. Graphical models. *STATISTICAL SCIENCE*, 19, 1 (2004), 140–155. (cited on page 21)

Ka Ki Ng, E. J. D., 2010. Object tracking initialization using automatic moving object detection. doi:10.1117/12.839126. https://doi.org/10.1117/12.839126. (cited on page 7)

Kalal, Z.; Mikolajczyk, K.; and Matas, J., 2012. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 7 (Jul. 2012), 1409–1422. doi:10.1109/TPAMI.2011.239. (cited on page 6)

Khémiri, A.; Kacem, A.; and BelaÃŕd, A., 2014. Towards arabic handwritten word recognition via probabilistic graphical models. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, 678–683. (cited on page 24)

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Čehovin Zajc, L.; Vojir, T.; Häger, G.; Lukežič, A.; Eldesokey, A.; and Fernandez, G., a. In *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*. (cited on pages 11 and 12)

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Čehovin Zajc, L.; Vojir, T.; Häger, G.; Lukežič, A.; and Fernandez, G., 2016a. The visual object tracking VOT2016 challenge results. http://www.springer.com/gp/book/9783319488806. (cited on pages 11 and 12)

Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebehay, G.; and Pflugfelder, R., b. The visual object tracking VOT2015 challenge results. In *2015 IEEE International Conference on Computer Vision Workshops (ICCVW)*. (cited on pages 11 and 12)

Kristan, M.; Matas, J.; Leonardis, A.; Vojir, T.; Pflugfelder, R.; Fernandez, G.; Nebehay, G.; Porikli, F.; and Čehovin, L., 2016b. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, 38, 11 (Nov 2016), 2137–2155. doi: 10.1109/TPAMI.2016.2516982. (cited on pages 6, 10, 11, and 57)

Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Porikli, F.; Čehovin Zajc, L.; Nebehay, G.; Fernandez, G.; Vojir, T.; Gatt, A.; Khajenezhad, A.; Salahledin, A.; Soltani-Farani, A.; Zarezade, A.; Petrosino, A.; Milton, A.; Bozorgtabar, B.; Li, B.; Chan, C. S.; Heng, C.; Ward, D.; Kearney, D.; Monekosso, D.; Karaimer, H. C.; Rabiee, H. R.; Zhu, J.; Gao, J.; Xiao, J.; Zhang, J.; Xing, J.; Huang, K.; Lebeda, K.; Cao, L.; Maresca, M. E.; Lim, M. K.; Helw, M. E.; Felsberg, M.; Remagnino, P.; Bowden, R.; Goecke, R.; Stolkin, R.; Lim, S. Y.; Maher, S.; Poullot, S.; Wong, S.; Satoh, S.; Chen, W.; Hu, W.; Zhang, X.; Li, Y.; and Niu, Z., c. The visual object tracking VOT2013 challenge results. In *2013 IEEE International Conference on Computer Vision Workshops (ICCVW)*. (cited on pages 11 and 12)

Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Čehovin Zajc, L.; Nebehay, G.; Vojir, T.; Fernandez, G.; Lukežič, A.; Dimitriev, A.; Petrosino, A.; Saffari, A.; Li, B.; Han, B.; Heng, C.; Garcia, C.; Pangeršič, D.; Häger, G.; Khan, F. S.; Oven, F.; Possegger, H.; Bischof, H.; Nam, H.; Zhu, J.; Li, J.; Choi, J. Y.; Choi, J.-W.; ao F. Henriques, J.; van de Weijer, J.; Batista, J.; Lebeda, K.; Öfjäll, K.; Yi, K. M.; Qin, L.; Wen, L.; Maresca, M. E.; Danelljan, M.; Felsberg, M.; Cheng, M.-M.; Torr, P.; Huang, Q.; Bowden, R.; Hare, S.; Lim, S. Y.; Hong, S.; Liao, S.; Hadfield, S.; Li, S. Z.; Duffner, S.; Golodetz, S.; Mauthner, T.; Vineet, V.; Lin, W.; Li, Y.; Qi, Y.; Lei, Z.; and Niu, Z., 2014. The visual object tracking VOT2014 challenge results. In *2014 European Conference on Computer Vision Visual Object Tracking Challenge Workshop*, vol. 8926 of *Lecture Notes in Computer Science*, 191–217. Springer International Publishing. doi: 10.1007/978-3-319-16181-5_14. (cited on pages 11 and 12)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (Eds. F. Pereira; C. J. C. Burges; L. Bottou; and K. Q. Weinberger), 1097–1105. Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. (cited on pages 25, 26, 27, and 28)

Kuo, C. and Nevatia, R., 2011. How does person identity recognition help multi-person tracking? In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1217–1224. doi:10.1109/CVPR.2011.5995384. (cited on page 14)

Kwon, J. and Lee, K. M., 2011. Tracking by sampling trackers. In *2011 IEEE International Conference on Computer Vision (ICCV)*, 1195–1202. doi:10.1109/ICCV.2011.6126369. (cited on page 10)

Lauritzen, S. L., 1996. *Graphical Models*. Oxford University Press. (cited on page 21)

Leal-Taixé, L.; Milan, A.; ; Reid, I.; Roth, S.; and Schindler, K., 2015. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, (Mar. 2015). https://arxiv.org/abs/1504.01942. ArXiv: 1504.01942. (cited on page 57)

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P., 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 2278–2324. (cited on pages 25 and 27)

Lenz, P.; Geiger, A.; and Urtasun, R., 2015. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In *2015 IEEE International Conference on Computer Vision (ICCV)*. (cited on page 15)

Leykin, A. and Hammoud, R., 2010. Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications*, 21, 4 (2010), 587–595. doi:10.1007/s00138-008-0176-5. http://dx.doi.org/10.1007/s00138-008-0176-5. (cited on pages 1 and 31)

Li, X.; Hu, W.; Shen, C.; Zhang, Z.; Dick, A.; and van den Hengel, A., 2013. A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, 4, 4 (2013). (cited on page 7)

Li, X.; Wang, K.; Wang, W.; and Li, Y., 2010. A multiple object tracking method using kalman filter. In *IEEE International Conference on Information and Automation*, 1862–1866. doi:10.1109/ICINFA.2010.5512258. (cited on page 16)

Li, Y.; Huang, C.; and Nevatia, R., 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2953–2960. doi:10.1109/CVPRW.2009.5206735. (cited on page 16)

Liang, G.; Lan, X.; Wang, J.; Wang, J.; and Zheng, N., 2017. A limb-based graphical model for human pose estimation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (2017), 1–13. doi:10.1109/TSMC.2016.2639788. (cited on page 24)

Liu, C.; Yao, R.; Rezatofighi, S. H.; Reid, I.; and Shi, Q., 2017a. Multi-object model-free tracking with joint appearance and motion inference. In *2017 International*

Conference on Digital Image Computing: Techniques and Applications (DICTA). doi: 10.1109/DICTA.2017.8227468. (cited on page 2)

Liu, F.; Lin, G.; Qiao, R.; and Shen, C., 2017b. Structured learning of tree potentials in CRF for image segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, (2017). (cited on page 21)

Liu, Y.; Li, H.; and Chen, Y. Q., 2012. Automatic tracking of a large number of moving targets in 3d. In *2012 European Conference on Computer Vision (ECCV)*, 730–742. Springer-Verlag. doi:10.1007/978-3-642-33765-9_52. http://dx.doi.org/10.1007/978-3-642-33765-9_52. (cited on page 16)

Lucchi, A.; Li, Y.; Smith, K.; and Fua, P., 2012. Structured image segmentation using kernelized features. In *2012 European Conference on Computer Vision (ECCV)*. (cited on page 21)

Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Zhao, X.; and Kim, T.-K., 2017. Multiple object tracking: A literature review. *arXiv preprint arXiv:1409.7618v4, 2018. URL: arXiv:1409.7618v4*, (2017). (cited on pages 13, 14, and 16)

Ma, C.; Huang, J.-B.; Yang, X.; and Yang, M.-H., 2015a. Hierarchical convolutional features for visual tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 3074–3082. (cited on pages 6, 7, 8, 29, 39, 45, 51, 54, and 57)

Ma, C.; Yang, X.; Zhang, C.; and Yang, M., 2015b. Long-term correlation tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5388–5396. doi:10.1109/CVPR.2015.7299177. (cited on page 6)

Matthews, I.; Ishikawa, T.; and Baker, S., 2004. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 6 (2004), 810–815. (cited on page 9)

Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; and Schindler, K., 2016. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, (Mar. 2016). http://arxiv.org/abs/1603.00831. ArXiv: 1603.00831. (cited on pages 1, 16, 17, 32, 40, 57, and 59)

Milan, A.; Rezatofighi, S. H.; Dick, A.; Reid, I.; and Schindler, K., 2017. Online multi-target tracking using recurrent neural networks. In *AAAI Conference on Artificial Intelligence*. (cited on page 29)

Milan, A.; Roth, S.; and Schindler, K., 2014. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 1 (Jan 2014), 58–72. doi:10.1109/TPAMI.2013.103. (cited on pages 1 and 14)

Milan, A.; Schindler, K.; and Roth, S., 2013. Detection- and trajectory-level exclusion in multiple object tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 15)

Minka, T., 2005. Discriminative models, not discriminative training. Technical report. https://www.microsoft.com/en-us/research/publication/discriminative-models-not-discriminative-training/. (cited on page 8)

Mitzel, D. and Leibe, B. In *2011 IEEE International Conference on Computer Vision Workshops (ICCVW)*. (cited on pages 14 and 16)

Moudgil, A. and Gandhi, V., 2018. Long-term visual object tracking benchmark. *arXiv preprint arXiv:1712.01358v3, 2018. URL: arXiv:1712.01358v3*, (2018). (cited on page 6)

Okuma, K.; Taleghani, A.; Freitas, N. D.; Freitas, O. D.; Little, J. J.; and Lowe, D. G., 2004. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision (ECCV)*, 28–39. (cited on page 1)

Pan, X.; Shi, J.; Luo, P.; Wang, X.; and Tang, X., 2018. Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI Conference on Artificial Intelligence*. (cited on page 29)

Pang, Y. and Ling, H., 2013. Finding the best from the second bests - inhibiting subjective bias in evaluation of visual tracking algorithms. In *2013 IEEE International Conference on Computer Vision (ICCV)*. (cited on page 10)

Papageorgiou, C. P.; Oren, M.; and Poggio, T., 1998. A general framework for object detection. In *1998 IEEE International Conference on Computer Vision (ICCV)*, 555–. IEEE Computer Society. http://dl.acm.org/citation.cfm?id=938978.939174. (cited on page 8)

Parkhi, O. M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A., 2014. A compact and discriminative face track descriptor. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. (cited on pages 1 and 31)

Patino, L.; Cane, T.; Vallee, A.; and Ferryman, J., 2016. Pets 2016: Dataset and challenge. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1240–1247. doi:10.1109/CVPRW.2016.157. (cited on page 16)

Pearl, J., 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 0-934613-73-7. (cited on page 21)

Pellegrini, S.; Ess, A.; Schindler, K.; and van Gool, L., 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE International Conference on Computer Vision (ICCV)*, 261–268. doi:10.1109/ICCV.2009.5459260. (cited on page 15)

Pfister, T.; Charles, J.; and Zisserman, A., 2015. Flowing convnets for human pose estimation in videos. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1913–1921. doi:10.1109/ICCV.2015.222. (cited on page 13)

Platt, J. C., 1999. Advances in kernel methods. chap. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, 185–208. MIT Press. ISBN 0-262-19416-3. http://dl.acm.org/citation.cfm?id=299094.299105. (cited on page 37)

Porikli, F.; Tuzel, O.; and Meer, P., 2006. Covariance tracking using model update based on lie algebra. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 728–735. doi:10.1109/CVPR.2006.94. (cited on page 14)

Qin, Z. and Shelton, C. R., 2012. Improving multi-target tracking via social grouping. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1972–1978. IEEE Computer Society. (cited on page 15)

Qing Wang, W. X. M.-H. Y., Feng Chen, 2011. An experimental comparison of online object-tracking algorithms. vol. 8138, 8138 – 8138 – 11. doi:10.1117/12.895965. (cited on page 10)

Reid, D. B., 1979. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24 (1979), 843–854. (cited on page 1)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 29 and 51)

Rodriguez, M.; Sivic, J.; Laptev, I.; and Audibert, J., 2011. Data-driven crowd analysis in videos. In *2011 IEEE International Conference on Computer Vision (ICCV)*, 1235–1242. doi:10.1109/ICCV.2011.6126374. (cited on page 16)

Ross, D. A.; Lim, J.; Lin, R.-S.; and Yang, M.-H., 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77, 1-3 (May 2008), 125–141. doi:10.1007/s11263-007-0075-7. http://dx.doi.org/10.1007/s11263-007-0075-7. (cited on pages 8, 9, 31, and 51)

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J., 1986. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chap. Learning

Internal Representations by Error Propagation, 318–362. MIT Press. ISBN 0-262-68053-X. http://dl.acm.org/citation.cfm?id=104279.104293. (cited on page 28)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L., 2015a. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 3 (2015), 211–252. doi:10.1007/s11263-015-0816-y. (cited on page 27)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L., 2015b. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 3 (2015), 211–252. doi:10.1007/s11263-015-0816-y. (cited on page 53)

Salti, S.; Cavallaro, A.; and Stefano, L. D., 2012. Adaptive appearance modeling for video tracking: Survey and evaluation. *IEEE Transactions on Image Processing*, 21, 10 (Oct 2012), 4334–4348. doi:10.1109/TIP.2012.2206035. (cited on page 10)

Santner, J.; Leistner, C.; Saffari, A.; Pock, T.; and Bischof, H., 2010. Prost: Parallel robust online simple tracking. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 723–730. doi:10.1109/CVPR.2010.5540145. (cited on page 10)

Sarawagi, S. and Gupta, R., 2008. Accurate max-margin training for structured output spaces. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08 (Helsinki, Finland, 2008), 888–895. ACM. doi:10.1145/1390156.1390268. http://doi.acm.org/10.1145/1390156.1390268. (cited on page 20)

Schuhmacher, D.; Vo, B.; and Vo, B., 2008. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56, 8 (Aug 2008), 3447–3457. doi:10.1109/TSP.2008.920469. (cited on page 16)

Schwing, A. G.; Hazan, T.; Pollefeys, M.; and Urtasun, R., 2012. Efficient structured prediction for 3d indoor scene understanding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2815–2822. doi:10.1109/CVPR.2012.6248006. (cited on page 21)

Scovanner, P. and Tappen, M. F., 2009. Learning pedestrian dynamics from the real world. In *2009 IEEE International Conference on Computer Vision (ICCV)*, 381–388. doi:10.1109/ICCV.2009.5459224. (cited on page 15)

Sevilla-Lara, L. and Learned-Miller, E., 2012. Distribution fields for tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1910–1917. doi:10.1109/CVPR.2012.6247891. (cited on page 8)

Shu, G.; Dehghan, A.; Oreifej, O.; Hand, E.; and Shah, M., 2012. Part-based multiple-person tracking with partial occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1815–1821. doi:10.1109/CVPR.2012.6247879. (cited on page 15)

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556 (2014). (cited on pages 26, 27, 28, and 53)

Smeulders, A. W. M.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; and Shah, M., 2014. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2014). https://ivi.fnwi.uva.nl/isis/publications/2014/SmeuldersTPAMI2014. (cited on page 10)

Supancic, J. S., III and Ramanan, D., 2013. Self-paced learning for long-term tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 6)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. http://arxiv.org/abs/1409.4842. (cited on page 27)

Takala, V. and Pietikainen, M., 2007. Multi-object tracking using color, texture and motion. In *2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–7. doi:10.1109/CVPR.2007.383506. (cited on page 8)

Tang, S.; Andres, B.; Andriluka, M.; and Schiele, B., 2015. Subgraph decomposition for multi-target tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5033–5041. doi:10.1109/CVPR.2015.7299138. (cited on page 15)

Tsochantaridis, I.; Joachims, T.; Hofmann, T.; and Altun, Y., 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6, 1 (September 2005), 1453–1484. (cited on pages 2, 20, 32, 33, 52, and 56)

Valmadre, J.; Bertinetto, L.; Henriques, J. F.; Tao, R.; Vedaldi, A.; Smeulders, A. W. M.; Torr, P. H. S.; and Gavves, E., 2018. Long-term tracking in the wild: A benchmark. *arXiv preprint arXiv:1803.09502v3, 2018. URL: arXiv:1803.09502v3*, (2018). (cited on page 6)

van de Weijer, J.; Schmid, C.; Verbeek, J.; and Larlus, D., 2009a. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18, 7 (Jul. 2009), 1512–1523. doi:10.1109/TIP.2009.2019809. http://dx.doi.org/10.1109/TIP.2009.2019809. (cited on page 8)

van de Weijer, J.; Schmid, C.; Verbeek, J.; and Larlus, D., 2009b. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18, 7 (Jul. 2009), 1512–1523. doi:10.1109/TIP.2009.2019809. http://dx.doi.org/10.1109/TIP.2009.2019809. (cited on page 58)

Vermaak, J.; Doucet, A.; and Perez, P., 2003. Maintaining multimodality through mixture tracking. In *2003 IEEE International Conference on Computer Vision (ICCV)*, 1110–1116 vol.2. doi:10.1109/ICCV.2003.1238473. (cited on page 1)

Viola, P. and Jones, M. J., 2004. Robust real-time face detection. *International Journal of Computer Vision*, 57, 2 (May 2004), 137–154. doi:10.1023/B:VISI.0000013087.49260.fb. https://doi.org/10.1023/B:VISI.0000013087.49260.fb. (cited on page 5)

Wang, N.; Shi, J.; Yeung, D.-Y.; and Jia, J., 2015. Understanding and diagnosing visual tracking systems. In *2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, 3101–3109. IEEE Computer Society. doi:10.1109/ICCV.2015.355. http://dx.doi.org/10.1109/ICCV.2015.355. (cited on pages 7 and 51)

Wang, Y.; Luo, X.; and Hu, S., 2017. Context multi-task visual object tracking via guided filter. In *2017 IEEE International Conference on Image Processing (ICIP)*, 4332–4336. doi:10.1109/ICIP.2017.8297100. (cited on pages 9 and 10)

Wei, Y.; Sun, J.; Tang, X.; Shum, H.-Y.; and Shum, H.-Y., 2007. Interactive offline tracking for color objects. In *2007 IEEE International Conference on Computer Vision (ICCV)*, 1–8. (cited on page 6)

Weng, S.-K.; Kuo, C.-M.; and Tu, S.-K., 2006. Video object tracking using adaptive kalman filter. *J. Vis. Comun. Image Represent.*, 17, 6 (Dec. 2006), 1190–1208. doi:10.1016/j.jvcir.2006.03.004. http://dx.doi.org/10.1016/j.jvcir.2006.03.004. (cited on page 9)

Weston, J. and Watkins, C., 1998. Multi-class support vector machines. (cited on page 19)

Wu, B. and Nevatia, R., 2006. Tracking of multiple, partially occluded humans based on static body part detection. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 951–958. doi:10.1109/CVPR.2006.312. (cited on page 16)

Wu, H.; Sankaranarayanan, A. C.; and Chellappa, R., 2010. Online empirical evaluation of tracking algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 8 (Aug 2010), 1443–1458. doi:10.1109/TPAMI.2009.135. (cited on page 10)

Wu, Y.; Lim, J.; and Yang, M., 2015. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 9 (Sept 2015), 1834–1848. doi:10.1109/TPAMI.2014.2388226. (cited on pages 6 and 10)

Wu, Y.; Lim, J.; and Yang, M. H., 2013. Online object tracking: A benchmark. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2411–2418. (cited on pages 5, 10, 11, 12, 40, and 59)

Wu, Z.; Kunz, T. H.; and Betke, M., 2011. Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1185–1192. doi:10.1109/CVPR.2011.5995515. (cited on page 15)

Wu, Z.; Thangali, A.; Sclaroff, S.; and Betke, M., 2012. Coupling detection and data association for multiple object tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1948–1955. doi:10.1109/CVPR.2012.6247896. (cited on page 15)

Xie, Y.; Zhang, W.; Li, C.; Lin, S.; Qu, Y.; and Zhang, Y., 2014. Discriminative object tracking via sparse representation and online dictionary learning. *IEEE Transactions on Cybernetics*, 44, 4 (April 2014), 539–553. doi:10.1109/TCYB.2013.2259230. (cited on page 8)

Yang, B. and Nevatia, R., 2012a. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1918–1925. doi:10.1109/CVPR.2012.6247892. (cited on page 15)

Yang, B. and Nevatia, R., 2012b. An online learned crf model for multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2034–2041. doi:10.1109/CVPR.2012.6247907. (cited on pages 14 and 15)

Zhan, B.; Monekosso, D. N.; Remagnino, P.; Velastin, S. A.; and Xu, L.-Q., 2008. Crowd analysis: a survey. *Machine Vision and Applications*, 19, 5 (Oct 2008), 345–357. doi:10.1007/s00138-008-0132-4. https://doi.org/10.1007/s00138-008-0132-4. (cited on page 15)

Zhang, J.; Ma, S.; and Sclaroff, S., 2014. Meem: Robust tracking via multiple experts using entropy minimization. In *2014 European Conference on Computer Vision (ECCV)*, 188–203. Springer International Publishing, Cham. (cited on page 9)

Zhang, L. and van der Maaten, L., 2013. Structure preserving object tracking. *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2013), 1838–1845. (cited on pages 13 and 21)

Zhang, L. and van der Maaten, L., 2014. Preserving structure in model-free tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 4 (2014). (cited on pages 2, 13, 32, 34, 38, 39, 45, and 57)

Zhang, L.; Zeng, Z.; and Ji, Q., 2011. Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation. *IEEE Transactions on Image Processing*, 20 (2011), 2401–2413. (cited on page 24)

Zhang, S.; Yao, H.; Sun, X.; and Lu, X., 2013a. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46, 7 (2013), 1772 – 1788. doi:https://doi.org/10.1016/j.patcog.2012.10.006. (cited on page 7)

Zhang, Z.; Shi, Q.; Zhang, Y.; Shen, C.; and Hengel, A., 2013b. Constraint reduction using marginal polytope diagrams for map lp relaxations. *arXiv preprint arXiv:1312.4637, 2013. URL: http://arxiv.org/abs/1312.4637*, (2013). (cited on page 34)

Zhao, Q.; Yang, Z.; and Tao, H., 2010. Differential earth moverâĂŹs distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 2 (2010), 274–287. (cited on page 51)