Titre: Title:	Multiple Object Tracking in Videos
Auteur: Author:	Zhuofei Kang
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Kang, Z. (2021). Multiple Object Tracking in Videos [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/6260/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Multiple Object Tracking in Videos

ZHUOFEI KANG
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Avril 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Multiple Object Tracking in Videos

présenté par **Zhuofei KANG**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Michel DESMARAIS, président
Guillaume-Alexandre BILODEAU, membre et directeur de recherche
Benjamin DE LEENER, membre

DEDICATION

Thanks to my parents,
for supporting me to receive the best education possible...
Thanks to my supervisor,
for academic guidance and generous help...

ACKNOWLEDGEMENTS

Thanks to my supervisor Guillaume-Alexandre Bilodeau, for helping me patiently during my studies and he was the guiding light for me to complete my master thesis. I would like to thank the members of the LITIV lab for providing me generous help during my studies.

RÉSUMÉ

Dans cette thèse, nous avons travaillé à l'amélioration des performances du suivi multiobjet. Plus précisément, nous avons étudié la conception d'une méthode pour mieux gérer les détections manquantes et les occlusions. En prenant comme base la méthode de suivi, IOU-tracker, nous avons ajouté à celle-ci un filtre de Kalman et des caractéristiques d'apparence Re-ID pour améliorer les correspondances entre les trajectoires et les objets. La composante du filtre de Kalman est utilisée pour prédire la position des rectangles englobants des objets lorsque des occlusions apparaissent. Et la composante des caractéristiques Re-ID contribue à faire correspondre les objets en mesurant la distance entre des caractéristiques d'apparence visuelle pour déterminer si elles représentent le même objet, même si les objets sont éloignés spatialement dans les trames. Ces composantes ont amélioré la robustesse et la précision de IOU-tracker.

Nous avons testé et évalué notre méthode sur l'ensemble de données UA-DETRAC. Cet ensemble de données inclut des vidéos de scènes de trafic automobile filmées dans divers scénarios. Nous démontrons que notre méthode surpasse la méthode de base, IOU-tracker, par une marge très significative et que nous obtenons des résultats surpassant l'état de l'art sur l'ensemble de données pour la métrique PR-MOTP. Dans l'étude d'ablation, nous avons noté que même si le filtre de Kalman aide le suivi, la plus grande contribution vient de la combinaison avec les caractéristiques de Re-ID. Dans les travaux futurs, nous considérerons la distance physique des objets dans l'image pour diminuer le coût de calcul et l'espace requis pour stocker les caractéristiques de ré-identification.

ABSTRACT

In this thesis, we worked on improving the performance of multiple object tracking. Specifically, we aimed to design a method to deal better with missing detections and occlusions. Based on the IOU tracker, we added a Kalman filter and Re-ID appearance features to match the trajectories and the objects. The Kalman filter component is used to predict the position of the bounding boxes of objects when occlusions appear. And the Re-ID feature component contributes to matching the objects by measuring the distance of the features to determine whether they represent the same object, even if the objects are distant spatially. These components improved the robustness and accuracy of the IOU tracker.

We tested and evaluated our model on the UA-DETRAC dataset. This dataset includes videos of traffic scenes in various scenarios. Our method is shown to outperform our baseline, IOU-tracker, by a very significant margin and getting state-of-art results on the dataset with the PR-MOTP metric. In the ablation study, we noted that even though the Kalman filter helps, the biggest contribution comes from combining the Re-ID feature with it. In future work, we will consider the physical distance of the objects in the image to decrease the computation cost and memory space necessary for the Re-ID features.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

The main goal of multiple object tracking (MOT) is simultaneously locating multiple objects in a given video and maintaining their identities, as well as recording their trajectories. Different from object detection algorithms, the MOT algorithms associate the object identity with each bounding box in order to distinguish the objects between intra-class instances, rather than only the rectangular bounding boxes identified by their coordinates, height, and width. Figure 1.1 illustrates the output of a MOT algorithm. Multiple object tracking plays an increasingly important role in computer vision and in industrial applications. The objects in MOT could be, for example, vehicles on a road, pedestrians on a street, and moving animals in a park or sport players in a playing field.

1.1 Definition of tracking

Compared with visual object tracking that tracks a single object, the research in multiple object tracking is more challenging because it involves several interacting steps and the datasets are time consuming to build. Because it is difficult to achieve in scenarios involving many objects, multiple object tracking is a very challenging problem. The multiple object tracker not only needs to distinguish the objects of interest from the background but it also needs to discriminate the different interacting objects based on their appearance so as to maintain the correctness of each object identity for a long time. Taking a detection-based multiple object tracking method as an example, when using the data association model to calculate the affinity between the object historical trajectory and object candidates in the current frame, the discriminative power of the tracked object feature models and the accuracy of the object candidates features directly affect the tracker performance.

Although it seems that VOT (Visual Object Tracking) and MOT (Multiple Object Tracking) are only different in the number of targets, the methods used to solve these are actually quite different. In terms of the object, visual object tracking algorithms are generally of unlimited types, while in multiple object tracking algorithms, objects are generally only from specific categories. In terms of duration, visual object tracking is more targeted at short-time video sequences, while multiple object tracking generally deals with longer videos, which involve the entrance, occlusion and departure of each target. From the perspective of implementation, visual object tracking focuses more on how to relocate the target, while multiple object tracking methods often focus more on how to match the objects based on detection results.

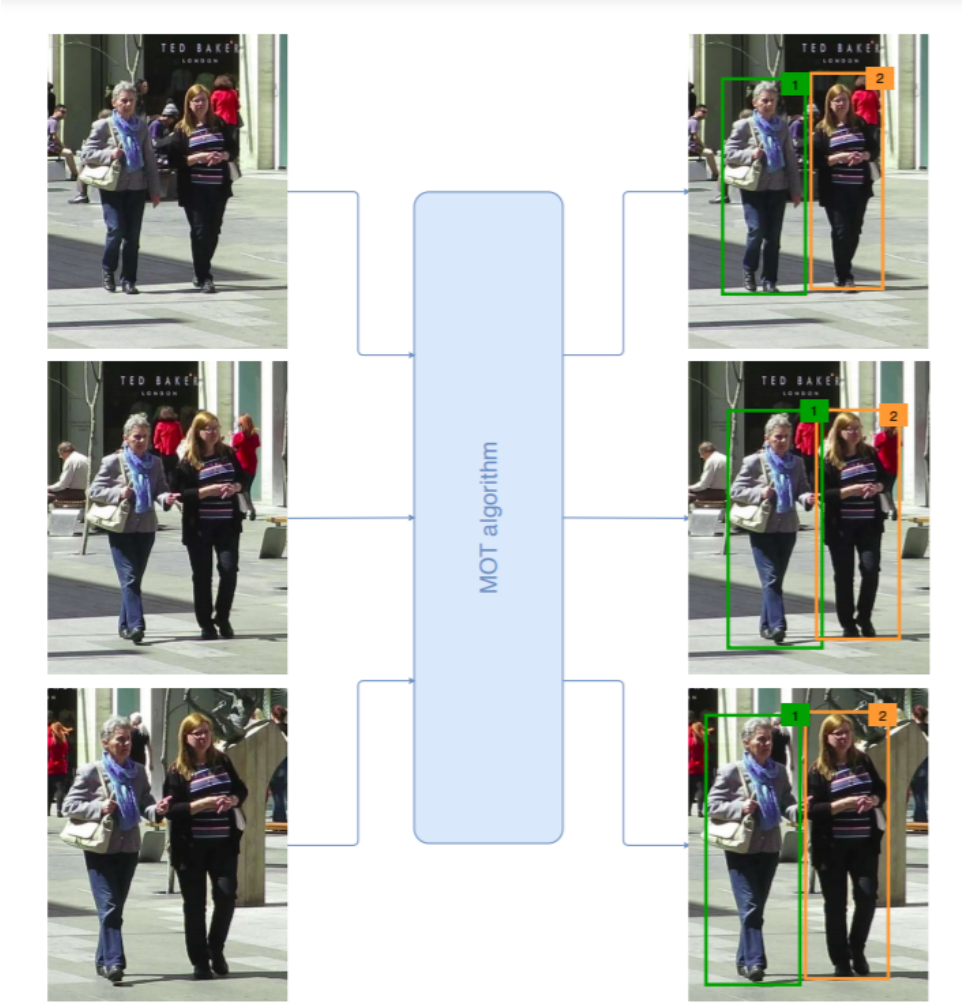


Figure 1.1 An illustration of the output of a MOT algorithm. The numbers above the bounding boxes represent the object identity [3](©2020, IEEE).

In recent years, object tracking algorithms based on deep learning have made great progress. Relatively speaking, the application of deep learning in the field of multiple object tracking is more limited to the learning of matching metrics and for obtaining detections. Indeed, progress in image recognition, such as image classification, detection and pedestrian re-identification problems can be directly applied to multiple object tracking problem. However, to consider the interactions between objects and the complexity of the tracking scenarios, the application of deep learning algorithms in the multiple object tracking problem is far from being fully studied.

Presently most multiple object tracking algorithms are based on detection algorithms, and can be classified in the detection-based tracking (DBT) category. DBT requires an object

detector to first detect the object in each frame of the video, and then each object is tracked separately by data association. It can automatically discover new objects and automatically terminate disappearing objects. Figure 1.2 presents the structure of detection-based tracking (DBT).

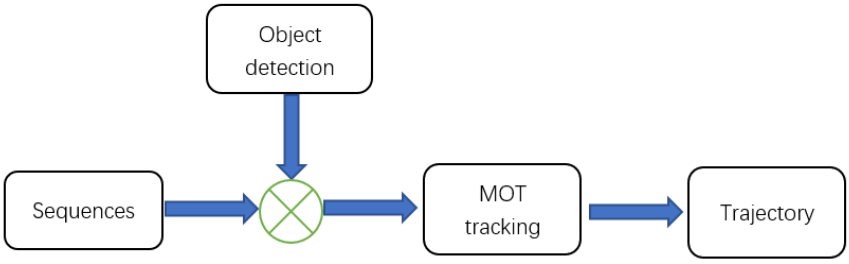


Figure 1.2 The structure of detection-based tracking (DBT).

Multiple object tracking algorithms can be divided into online tracking and offline tracking. Figure 1.3 presents the structure of online tracking and of offline tracking. In online tracking, the image sequence is handled step by step. The online tracking requires that when processing each frame, only the information in the current frame and in the previous frames can be used to determine the tracking result of the current frame, and the tracking result of the previous frames cannot be modified according to the information of the current frame. Based on the information from a cost matrix, trajectories are produced on the fly. Offline tracking allows the use of information from subsequent frames to obtain a globally optimal solution. The setting of offline tracking is not very suitable for practical application scenarios, but offline tracking in the form of a "batch" is feasible, but it will cause a little delay. Offline tracking can use a batch of frames to perform the data association, which can determine the tracking result of the current frame by the information from previous frames and future frames.

The online multiple object tracking methods are usually faster, but it is worth noting that online multiple object tracking is not equivalent to real-time tracking: online multiple object tracking emphasizes tracking frame-by-frame, processing online input of video frames. Real-time tracking means that the overall running speed of the tracking system reaches a certain order of magnitude (for example, the tracking speed is greater than 25 frames per second). Because offline multiple object tracking methods require future information, offline multiple object tracking methods cannot be applied in real-time tracking, while online multiple object tracking methods can. At the same time, because the online multiple object tracking methods do not use the information of future frames, in complex scenes (such as frequent occlusion of the target, etc.), the tracking performance of this type of method is generally not as good as

offline multiple object tracking methods. Online multiple object tracking methods are more prone to problems, such as tracking drift and wrong identity switching, which is also the research focus and difficulty of these methodes

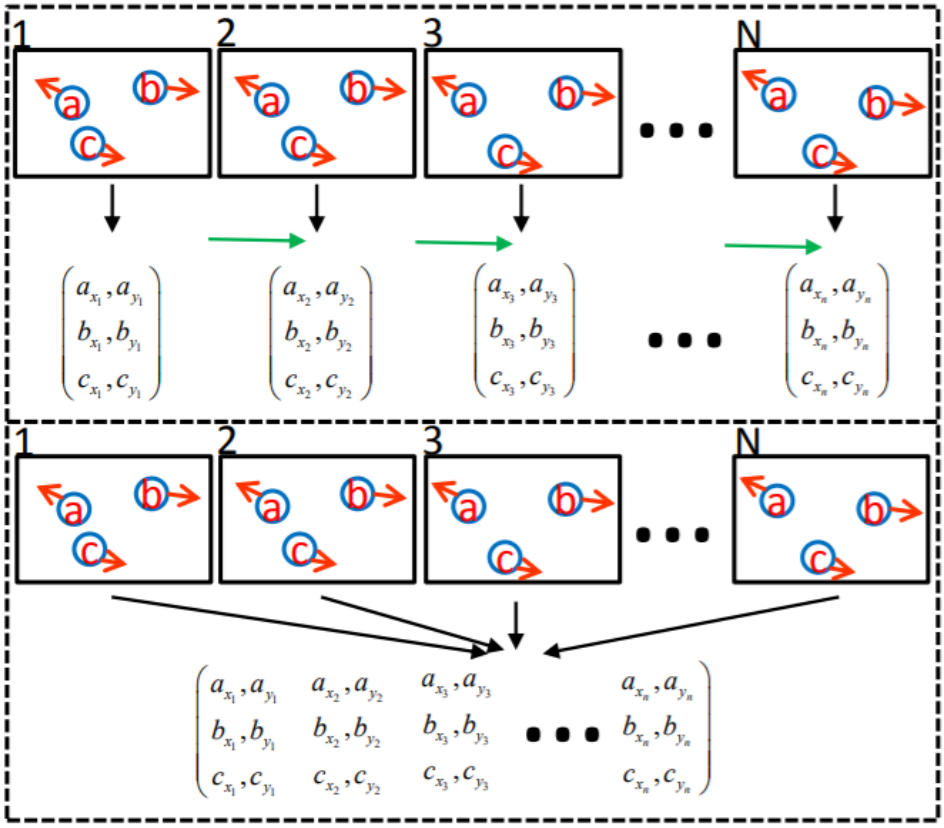


Figure 1.3 The structure of online tracking and offline tracking [4]. The top one shows the structure of online tracking and the bottom one presents the structure of offline tracking. The circles represent different objects a, b and c, the green arrows represents tracking results from past frames and the matrices present the location of the objects.

1.2 Challenges in tracking

Most of the previous multiple object tracking algorithms can only be applied to specific scenarios. To achieve a wider applicability and strong robustness, there are still challenging problems to be solved. At present, the challenging factors in multiple object tracking are:

- the occlusion problem: During the movement of the objects, other objects in the video may block the view of target objects. For example, in the case of road congestion, partial or even full occlusion occurs between vehicles. When the object is partially

occluded, the object features and shape information will be partially lost. At the same time, the shape and feature information of the occluder are introduced, so when using feature tracking, there will be unreliable tracking and tracking model drift. When the object is completely occluded, the object features disappear. At this time, only the prior knowledge of the moving object can be used for tracking.

- Changes in object scale: When the distance between the moving object and the camera increases or decreases, the objects will undergo scale changes in the image frame. Since the moving object and the camera are in relative motion, the relative size of the object in the image frame is also constantly changing. If the tracking bounding box cannot be resized during the object tracking process to adapt to changes in the object scale, some non-target feature information or only part of the target feature information will be included in the object model, which will lead to unstable tracking.

- Objects with similar appearance: During the multiple object tracking process, the extracted features cannot describe the appearance of the object distinctively, due to similar objects in the video. For example, the shape and color similarity of many vehicles are very high. In that case, identity switches are more likely to happen.

- Complex environment: In object tracking, the external environmental factors can be very complicated. Changes in weather (rain, snow, sand and fog) and camera shakes will blur the video frames, which greatly affects the tracking. The change in lighting is also an important factor that cannot be ignored in the tracking process. When the object passes in the shadow cast by buildings, trees, etc., some of the object feature information will change rapidly, which will cause tracking instabilities.

The above problems seriously affect the performance of multiple object tracking and greatly restrict the applicable scenarios of object tracking.

1.3 Research Objectives

In this thesis, we focus on improving the tracking performance in the case of traffic scenes. In particular, we want to design a method to deal better with missing detections and occlusions. In order to achieve this aim, we have several sub-objectives:

- To predict the position of the object in case of occlusions in the short term. We want to propose a method to track objects with occlusions;

- To match the trajectories with unmatched objects in the following frames. Given the fast motion of some objects, we want to include other features for matching objects between frames;
- To reconnect trajectories in case of occlusions or missing detections. We aim to propose a method to reconnect or resume interrupted trajectories.

1.4 Organization of the thesis

The rest of this thesis is structured as follows. Chapter 2 is a literature review on multiple object tracking and object detection. Chapter 3 presents our proposed method in detail. We introduce the baseline method and the changes we implemented. A detailed description of the proposed method is also given. Chapter 4 provides the experimental results as well as the analyses of the components and parameters. The last part is the conclusion of this thesis and the discussion of further research.

CHAPTER 2 LITERATURE REVIEW

This chapter has two parts. Section 2.1 presents multiple object tracking and the literature review of the main methods based on the tracking by detection paradigm, in particular, algorithms dealing with the data association problem. Section 2.2 describes the main object detection methods, which are the basis of object tracking.

2.1 Multiple Object Tracking

Compared to visual object tracking, the research on multiple object tracking (MOT) includes more challenges as stated in the introduction. The recent MOT methods are mostly based on the tracking by detection paradigm, focusing on the object representation (also called state) and the data association problem. The tracking by detection paradigm requires an object detector to detect the objects in each frame of the video, and then each object is matched from frame to frame. How to establish the object associations and how to identify the same object in the different frames are worthy of more attention.

SORT [17] is a typical example of the tracking by detection paradigm. Several works use a similar framework as the one of SORT. SORT has four basic components: object detection, object state prediction, data association and management of track identities. These are also the basic components of many MOT algorithms that follow the tracking by detection paradigm. In its original version, SORT uses Faster R-CNN [10] for object detection. Any other object detector can be used. For the state of a given object, SORT simply models the object using its center coordinates, its scale (area), the aspect ratio of its bounding box and their rates of change without using any pixel appearance information. The authors use the Hungarian assignment algorithm [18] for data association. The assignment cost matrix used is computed with the intersection over union (IOU) between the predicted bounding boxes of the currently tracked object for the current frame (based on observations in previous frames) and the bounding boxes of the objects from the detector in the current frame. If in a frame, a tracked object does not match with an object detected in that frame based on the IOU, the object is considered to have disappeared.

The SORT algorithm uses Kalman filtering [19] to do the prediction of object states frame-by-frame and the Hungarian algorithm to achieve the data association. The prediction of the objects is used to correct the current bounding box or to infer missing bounding boxes. The SORT algorithm achieves good performance at high frame rates. However, because SORT

ignores the precise visual appearance of the detected objects, it is accurate only when the uncertainty for the state estimation of the object is low. DeepSort [20] has been proposed to address this uncertainty problem in data association.

For data association, DeepSORT proposes an overall similarity metric, which is a combination of motion and appearance information to improve the performance of SORT, using Mahalanobis distance as the similarity measure for motion features and cosine distance as the similarity measure for appearance features. Furthermore, DeepSORT applies a CNN network to learn and extract features on large-scale pedestrian datasets, in order to increase the robustness against misses and occlusions with a strong appearance model. Finally, a cascading matching method is used by DeepSORT, so that the tracks with higher recent activity are preferentially matched during data association.

The IOU tracker [13] proposed by Bochinski et al. relies on the IOU between each object bounding box in adjacent frames, and can achieve a good performance if the frame rate is high enough and the detection quality is good enough. The IOU Tracker does not consider appearance information, does not predict the motion of objects, and does not use complex matching algorithms. It uses a greedy strategy to match all bounding boxes of objects between adjacent frames. Specifically, for each object being tracked, the detection with the largest IOU is selected as the best potential match. If the IOU of these two bounding boxes (object detection) is greater than a threshold, the two are considered a match, otherwise the matching fails. If a track fails to match with a new detection in several consecutive frames, the object is considered to have left the scene. It is kept or rejected as a finished track according to its duration of existence and confidence. Finally, if there is a detection bounding box that does not match any tracks of previous frames, it is considered as a new object that has appeared and a new track is created for it.

Bochinski et al. improved the IOU tracker by considering visual information, and proposed the V-IOU tracker [2]. When a track cannot match a bounding box in successive frames, it no longer simply considers that the object has left the field of view, but starts a visual object tracker at this time to try to continue tracking this target until a time-to-live threshold is reached without any new matches. When a new track is created, V-IOU starts a visual tracker and track backward in time for several frames to try to find the objects. If the new track and a terminated track can be matched by IOU with the help of the backward tracking, then they are merged.

Milan et al. proposed an end-to-end model learning approach for online MOT using Recurrent Neural Networks [5], which is an attempt at using deep learning in multiple object tracking. The advantages of using a RNN for MOT is that it is model-free, that is, there is no

need to learn some prior knowledge, such as target dynamics and clutter distribution. This method implemented all processes involved in multiple object tracking with deep learning, that is, prediction, state update, data association and management of tracks. The figure 2.1 shows a RNN-based model for state prediction, state update, and object existence probability estimation and a LSTM-based architecture for data association. The sequences of images can be considered as sequences of inputs. The RNN achieves the prediction and the update of single object state and the association with other targets, and then outputs the predicted trajectory. Unfortunately however, this end-to-end deep learning method could not work as well as expected because its data association and prediction are not as good as methods, like a Kalman filter and the Hungarian algorithm.

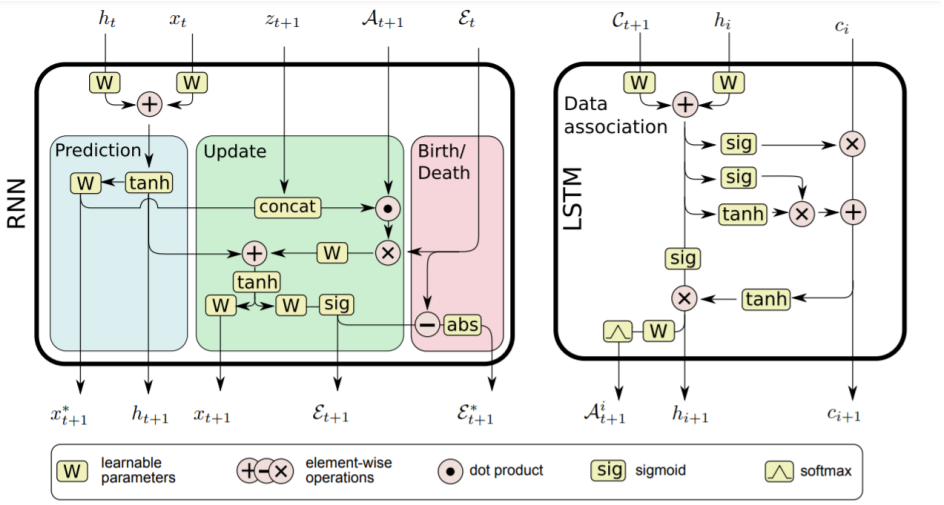


Figure 2.1 A RNN-based structure for state prediction, state update, and object existence probability estimation and A LSTM-based architecture for data association [5]. In this figure, x_t denotes the current state, h_t denotes hidden state of the RNN, σ_t denotes existence probabilities, A_{t+1} denotes the data association and z_{t+1} denotes the measurements in the following frame(©2017, IEEE).

Kieritz et al. proposed an online joint detection and multiple object tracking framework, considering that detection and tracking could benefit from each other [21]. Indeed, the similarity model used in tracking can reuse the appearance features calculated by the detector, and the detector can use object information in past frames to prevent missing object detections. The authors used SSD [22] as the detector, and input the detection results to a RNN network, where an association metric between detections and tracks is calculated, to achieve data association and to update the tracks in each frame. The Hungarian algorithm [18] is used to perform data association between the detections and the tracks, and then trajectories are updated in the tracking set. The association metric is also used to guide a modified

non-maximum suppression method to remove redundant detections.

Dual Matching Attention Networks method [23] proposed by Zhu et al. applies visual object trackers in MOT by integrating a single object tracker and data association. In order to solve the problem of extremely uneven distribution of positive samples (objects) and negative samples (background), a cost sensitive loss has been proposed, which makes the tracker focus on hard negative examples during training, so that it improves the robustness of the visual object tracker. In order to mitigate the problem of noise, occlusion and lack of proper data association, spatial attention has been implemented to improve the matching of image pairs. In order to adaptively assign different levels of attention to different samples in the trajectory to deal with the noisy environment, Dual Matching Attention Networks used a temporal attention network to filter out unreliable samples in the trajectory.

Xiang et al. proposed the Learning to Track algorithm [24] where the online multiple object tracking problem is integrated into a Markov Decision Processes (MDP). A MDP model is established for each goal to solve (object to match in data association). In MDP, learning a policy mainly involves learning the correlation of similar data. Policy learning mainly uses the method of reinforcement learning, which is more conducive to the correlation between offline data and online data. The method in this article regards the appearance and disappearance of targets as a state migration in MDP to complete. The learning of the MDP policy is done with visual object tracking. Then multiple MDPs are used for the MOT situation.

He et al. proposed a Tracking-by-Animation framework [25], which is a label-free and an end-to-end unsupervised learning of MOT. In the Tracking-by-Animation (TBA) framework, a differentiable neural model tracks objects from an input video and then animates these objects into reconstructed frames to predict their future positions, and then the reconstruction errors drive the learning by back propagation (object motion is learned for the reconstruction errors). Meanwhile, in order to avoid overfitting and disrupted tracking, the authors proposed a Reprioritized Attentive Tracking (RAT) module that improves the robustness of data association by learning to perform it. The Tracking-by-Animation framework is an unsupervised method, avoiding the expense of labeling data.

Most of existing MOT methods execute the object detection and data association modules separately, which may cause efficiency issues. Therefore, in order to save computation and improve the speed of tracking, Wang et al. proposed a near real-time MOT system, called Joint Detector and Embedding model (JDE) [6], where the detector and the embedded models are integrated into a single neural network. The figure 2.2 shows the network architecture of the Joint Detector and Embedding model. The appearance information used in data association does not need to be computed separately. The training process is modeled as a multi-task

learning problem, including classification, bounding box regression and embeddings learning. Joint learning in a single network improved the efficiency of tracking significantly. However, JDE does not work well when too many occlusions happen.

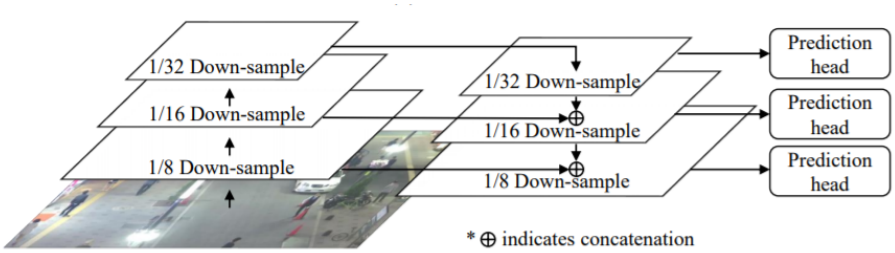


Figure 2.2 The network architecture of Joint Detector and Embedding model [6]. In the pyramid network, a prediction head is added at each resolution so that the training process can be considered as a multi-task learning problem. The heterogeneous losses can be learned by learning a set of auxiliary parameters from the task-dependent uncertainty(©2019, IEEE).

In order to overcome the temporal errors in multiple object tracking, Yoon et al. proposed a joint-input Siamese Network (JI-Net) considering the historical appearance matching [26]. When an occlusion happens or noise near an object exists, it can cause tracking failures or ID-switches. The authors proposed a novel matching method, historical appearance matching (HAM) based on a siamese network, that can avoid the matching ambiguity problem in the current frame. As well, due to the variable distribution of detection confidences depending on the scene, the authors proposed a scene adaptive detection filter, so that it removes the noise in detection results effectively. JI-Net has then been improved by proposing Deep Temporal Appearance Matching Association (Deep-TAMA) [27], where historical appearance matching is implemented for adaptive appearance modeling in the framework. Each matching score is associated based on the confidence of each historical appearance. Deep-TAMA can avoid ambiguities in the process of tracks-detections matching due to the likelihood calculation method based on a set of previous reliable templates of the target, but ID-switches still occur frequently.

Tang et al. viewed the multiple pedestrian tracking problem as a minimum cost lifted multicut problem (LMP), which is a novel graph-based clustering formulation [28]. Nodes are detections and edges are possible associations. Lifted edges in LMP introduce long-range information where nodes should be joined/cut. Long-term false connections are penalized by forcing valid paths in the feasible solution. In order to match over a long temporal gap, the authors proposed a novel re-identification model fusing detections of body part into a CNN that can be regarded as an attention module that focus on relevant parts.

The Deep Affinity Network (DAN) [29] is used to pair the extracted features of detected targets in any two frames to infer the correlation of the objects. According to the appearance model of the object and its affinity between different frames, an end-to-end deep learning approach is trained for data association. Multiple objects that appear and disappear between video frames are also considered by DAN. Using the DAN network, the affinity calculation for the video frames is performed, and the objects in the video frames are related to the objects in multiple previous frames. Then using the calculated affinity, the Hungarian algorithm is used to generate a reliable trajectory. Due to the affinity calculation in several frames, DAN works well when occlusion happens.

Bergmann et al. proposed to convert an object detector into a tracker, without specifically training the tracker on occlusion, as well as involving re-identification feature extraction and motion prediction to complete the tracking task [7]. No training nor optimization is required. The object detector bounding box regression is used to predict the new position of the object in the next frame, and it is extended through simple re-identification based on a Siamese network and camera motion compensation, demonstrating the scalability of the tracker. The method proposed by the authors can solve most simple tracking scenarios, and explores some challenges and problems of tracking. The figure 2.3 shows the processing steps of their tracker, called Tracktor, which accomplishes multiple object tracking only with an object detector.

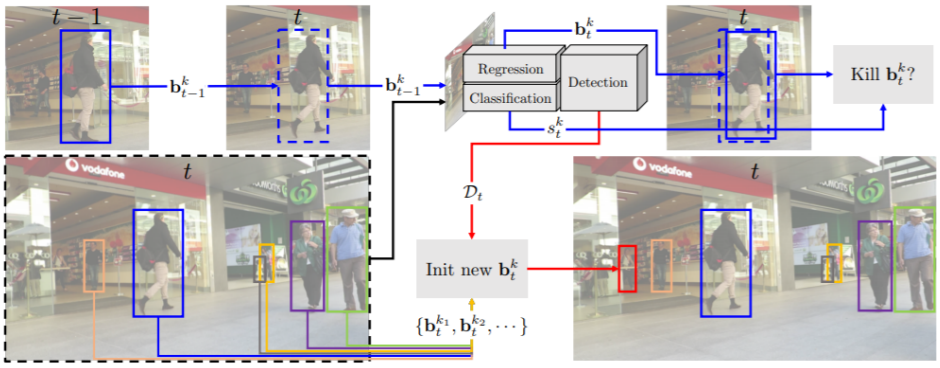


Figure 2.3 Tracktor accomplishing multiple object tracking only with an object detector [7]. The blue boxes and arrows mean the regression of the detector aligning the existing track bounding boxes b^k_{t-1} in the past frame $t-1$ and the position of the object in the current frame t. The red box and arrows mean the object detector initializing a set of new detections(©2019, IEEE).

Continuous Energy Minimization for Multi-Target Tracking [30] proposed by Milan et al. transforms the tracking problem into an energy minimization problem. In the solution space,

each solution corresponds to a cost or energy. Then, the cost function is minimized. In this work, they aim to find a suitable method to obtain the most optimal solution. In this paper, the solution space is all possible combinations of tracklets composed of detections. Each combination has a cost, and the method is looking for the optimal combination. It constructs a continuous cost function, which is easy to solve and it uses jump move to jump out of the local optimum and find the global optimum.

Zhang et al. proposed an end-to-end DNN tracking approach, Flow Fuse-Tracker (FFT) [31], which avoids solving the MOT problem in two separate steps, performing motion feature extraction and data association simultaneously. Target flowing and target fusing are the novel efficient techniques proposed in the FFT model. A FlowTracker DNN is applied to extract multiple target-wise motions from optical flow, and the objects proposed by FlowTracker and an object detector are refined and fused by the FuseTracker DNN module. Therefore, FFT can obtain the data association results in the tracking process directly. FFT reduces computational cost and it can be improved by better optical flow and object detector networks.

Methods based on graphs regained some attention in recent years, Braso et al. proposed a MOT solver based on the natural graph structure of the MOT problem to achieve feature learning and solution prediction [32]. The problem is formulated based on the classic minimization cost flow in MOT, and with a MOT approach based on a solver message passing networks with a novel time update step. This method shows that graph optimization method based on message passing can work in the field of multiple object tracking.

In order to avoid the computational cost in associating an indefinite number of objects between frames, Zhang et al. proposed a novel efficient Deep Neural Network, Frame-wise Motion and Appearance (FMA) [33], which achieves data association among an indefinite number of objects simultaneously. Frame-wise Motion Fields (FMF), learned by FMA, are used for matching the objects in different frames, and then Frame-wise Appearance Feature (FAF) are used to modify the incorrect matching. This method can achieve real-time multiple object tracking and the inference computation times of the networks will not increase with the number of objects.

Considering the Relation network for object detection [34] that was proposed by Dai et al., where encoded context information is used for object detection, Xu et al. proposed a joint framework to expand the object-object relation from the spatial domain to the space-time domain by using multiple clues to measure the similarity in an end-to-end manner [8]. The figure 2.4 shows the spatial-temporal relation networks computing similarity scores between tracklets and objects. The spatial domain inference is performed in each frame. This spatial inference process uses automatically learned topology information to enhance the appearance

characteristics of the objects. Then through spatial relationship reasoning, the enhanced features are aggregated on multiple frames. This algorithm works well on many MOT datasets.

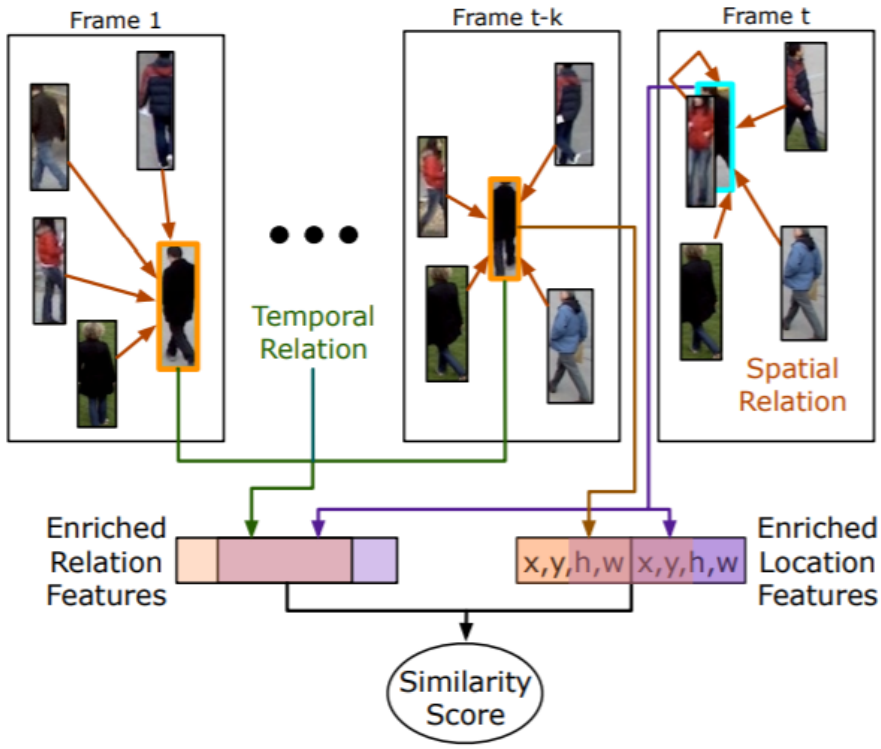


Figure 2.4 The spatial-temporal relation networks (STRN) to compute the similarity scores between tracklets and objects [8]. The orange bounding box and the blue bounding box indicate the same person localized with the use of topological information that allows tracking through occlusions(©2019, IEEE).

In order to improve the results of object tracking, Voigtlaender et al. hypothesized that the tracking should be at the pixel-level (in contrast to the bounding box-level) and it should be performed jointly considering object detection, tracking and segmentation [35]. They extended Mask R-CNN by 3D convolutions and proposed Track R-CNN, which combines temporal context information and generates a correlation vector for each detection through a correlation head. According to the similarity of the detected correlation vector, it expands or build new trajectories. The authors proposed also a novel measurement, soft Multi-Object Tracking and Segmentation Accuracy (sMOTSA) that can measure segmentation as well as detection and tracking quality.

Lee et al. proposed a new deep neural network architecture, Feature Pyramid Siamese Network (FPSN) [36], based on Feature Pyramid Network [12], improving the basic Siamese

Network. FPSN obtains the features to describe objects from multiple levels, therefore capturing sophisticated discriminative features. FPSN-MOT could achieve an outstanding tracking performance by not only considering the appearance information from FPSN but also the spatio-temporal information from the difference between detection and existing tracking.

2.2 Object detection

The performance of MOT is very dependent on the performance of the object detector used. The object detection algorithms can be divided into two categories: two-stage detection algorithm and one-stage detection algorithm. One-stage method can be further divided into bounding box-based and point-based. In general, a two-stage algorithm has an advantage in accuracy, and a one-stage algorithm has an advantage in speed.

A two-stage detection algorithm, as the name implies, divides the detection problem into two stages: first generating regions proposals, and then the second, classifying the candidate regions (generally, a position refinement is also required at this step). Most of the two-stage algorithms are based on a region proposal network to speed up the classification by processing only regions with high probability of containing an object. A one-stage detection algorithm does not require the region proposal stage and directly generates the class probability and position coordinate value of the objects. Because the final detection result can be directly obtained, the detection speed is faster. In this part, we will present some popular object detection methods.

2.2.1 One-stage detection algorithms

Redmon et al. presented the YOLO network [37], leading to the first one-stage detection approach. YOLO directly returns the position of the bounding boxes and the classes to which the bounding boxes belong to in the output layer (the whole image acts as an input to the network, transforming the problem of object detection into a regression problem). The coordinates of the bounding box, the confidence of the object contained in the box, and the class probabilities are obtained directly from all the pixels of the entire image. Since there is no complicated detection process, YOLO can complete the object detection task very quickly. However, at the time of publication, the detection accuracy of YOLO is lower than that of other state-of-the-art object detection methods, and also YOLO performs poorly on small objects.

Lin et al. [38] analyzed the reason why an one stage detector is not always good. They found that it is because of an extremely unbalanced positive and negative sample ratio and

of the gradient that is dominated by easy examples. They presented the focal loss which is an improved cross-entropy loss. The focal loss successfully solves the problem of extremely unbalanced positive and negative sample ratio at the time of target detection, and the problem of the target detection loss that is easily affected by large batches of negative samples.

Law et al. presented a novel method based on keypoints, CornerNet [39], independently predicting the position of the upper left and lower right corners of the object bounding box, and then gathering the corresponding upper left and upper right corners by associative embeddings to get an object instance in the image. The network has two branches, one for predicting top-left corners and the other for bottom-right corners. Each branch has three outputs, heatmaps predicting corner points, embeddings mainly predicting to which object the corner points belong to, and the last one modifying the position of the corner points for fine positioning of the bounding box. CornerNet used a corner pooling method to gather the detection results on the edge to the apex. But even with corner pooling, using only two vertices can lead to a lot of false detection. CenterNet [40] adds a center point prediction based on the same method as CornerNet. The requirement to form an object is not only the matching of two vertices, but also the center of the box defined by the two vertices. Due to the introduced center point, CenterNet alleviates a lot of false detection. CenterNet also proposed center pooling and cascade corner pooling to improve the accuracy of center point and vertex prediction.

Zhou et al. [9] proposed that each object can be represented as keypoint, thus the object detection problem can be considered as a keypoint detection problem. The image generates a heatmap after being fed into a fully convolutional network, and peaks in this heatmap represent object centers that are related to the bounding box of the objects, and then the image features corresponding to this point are used to directly predict the length and width of the bounding box. During inference, since there is only one point for each object, there will be no overlap of several prediction bounding boxes, and therefore, there is no need to use a NMS algorithm to remove duplication. Figure 2.5 shows the principle of object as point.

SpotNet [41] proposed by Perreault et al., improves object detection with points by adding an attention module, which focuses on the interesting area while decreasing the signal of non-relevant areas. This method adds an attention model to CenterNet (object as points) [9]. The foreground/background segmentation labels are obtained by background subtraction or optical flow, and then are used to produce foreground/background segmentation maps by training an object detection model on the labels. A self-attention mechanism is added into the network to weight the feature map using the learned segmentation maps to produce the bounding boxes. This method improves mAP significantly on UA-DETRAC and UAVDT

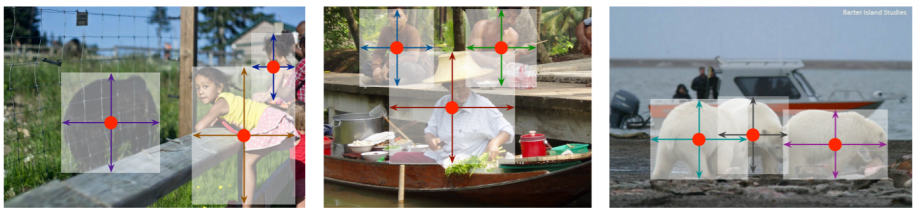


Figure 2.5 Each object can be represented by the center point of its bounding box. The length and width of the bounding box are inferred from the keypoint feature at the center [9](©2019, IEEE).

datasets.

2.2.2 Two-stage detection algorithm

Girshick et al. proposed the R-CNN algorithm [42], which is the first CNN-based object detection method, where region proposals and a CNN are used in object detection. An external method generates region proposal to obtain a list of candidate regions. For each candidate region, it is first normalized to the a specific size, and features are extracted using a deep neural network, and then the features are sent to a SVM classifier to determine whether the region belongs to a given class. Finally, the candidate bounding box position is refined by regression. This algorithm indeed improves the detection accuracy significantly and became a milestone in object detection.

In the work by Ren et al., Faster R-CNN [10] integrates feature extraction, proposal extraction, bounding box regression, and classification into a single network. In contrast to R-CNN, an external region proposal method is no longer needed. This greatly improved the overall performance, especially in terms of detection speed. Faster R-CNN introduces the Region Proposal Network (RPN), where the candidate box extraction is merged into the deep neural network, so that the detection speed is greatly improved because the same features are used for both generating region proposals and performing the classification. Therefore, neural network layers are shared for the two tasks. The figure 2.6 shows the structure of Region Proposal Network.

Dai et al. proposed the Region-based Fully Convolutional Networks (R-FCN) [11], which addresses the contradiction between translation-invariance in image classification and translation-variance in object detection, and uses position-sensitive score maps improving the detection speed as well as accuracy. The figure 2.7 shows the overall architecture of R-FCN. R-FCN is based on the framework of Faster R-CNN. Firstly, the backbone network is replaced by

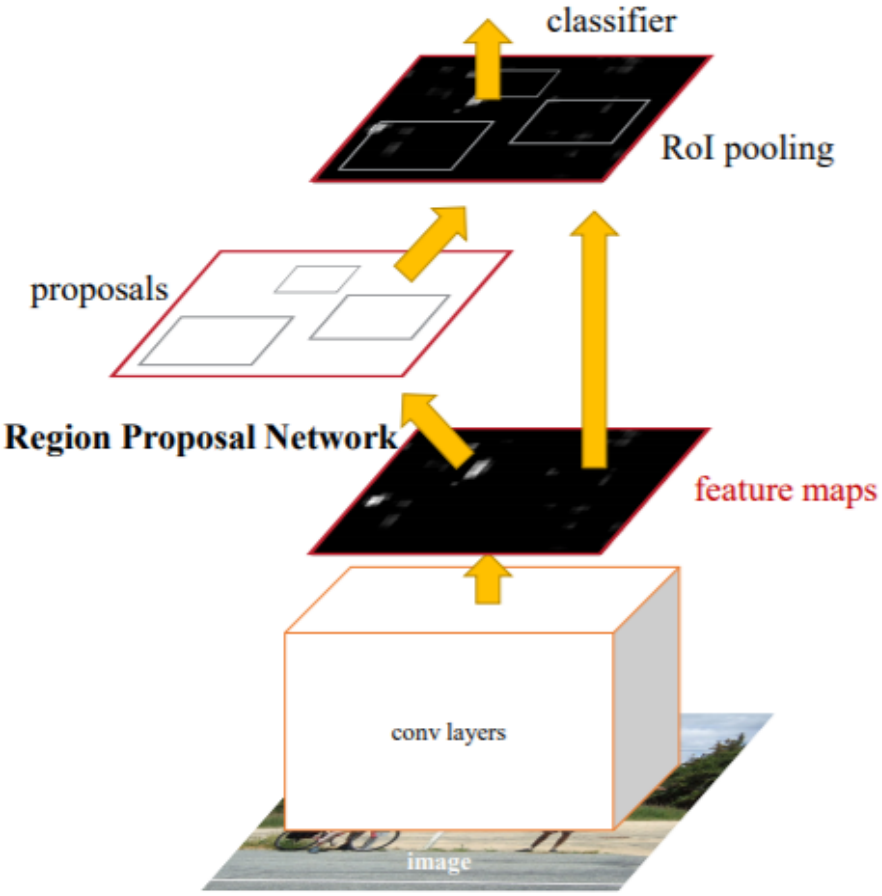


Figure 2.6 The structure of Faster RCNN [10]. The feature maps of the image are generated by Convolutional layers and then the region proposals are extracted by the region proposal network. The feature maps and proposals are feed in the ROI pooling to produce proposal feature maps. Finally, the proposal feature maps are used to calculate the category of the proposal, and bounding box regression is applied to obtain the final precise position of the object(©2015, IEEE).

ResNet-101. Secondly, the prediction of object bounding boxes is achieved by convolutions before ROI pooling. Since ROI pooling loses location information, location information is added before pooling, where different score maps are assigned to different locations for detecting targets. After the pooling, the score maps obtained from different locations can be combined to reproduce the original location information.

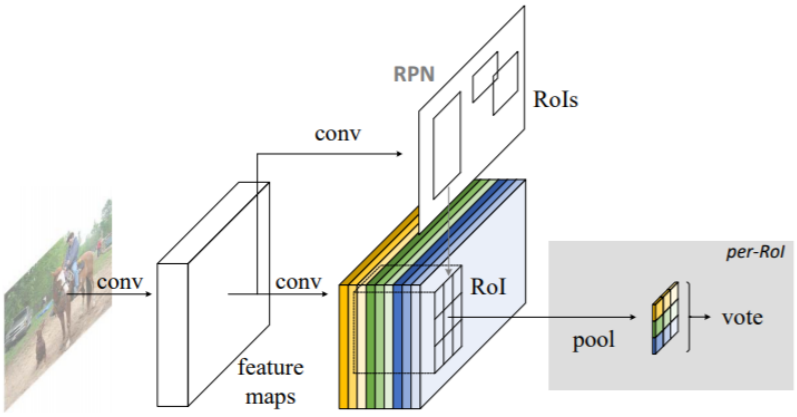


Figure 2.7 Overall architecture of R-FCN [11]. Feature maps are extracted by the Convolutional layers and then they are fed into a fully convolutional network generating position-sensitive score maps. The ROI pooling is performed on one of the k^2 maps, and then the final detection results are obtained through the vote operation (avg pooling or max pooling)(©2016, IEEE).

The Feature Pyramid Network [12] proposed by Lin et al. focuses on the multi-scale problem in object detection. Through a new network connection method, the performance for small object detection is greatly improved without substantially increasing the amount of calculation over the original model. The figure 2.8 shows the architecture of the Feature Pyramid Network. There are three connection lines in Feature Pyramid Networks, a bottom-up connection, a top-down connection, and lateral connections. The Feature Pyramid Network considers both high resolutions of low-level features and high semantic information of high-level features, achieving a better prediction by blending the features of these different layers. And the prediction is performed separately on each fused feature layer.

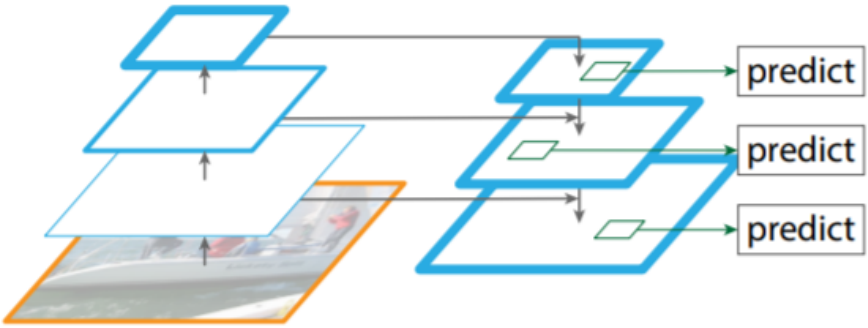


Figure 2.8 The architecture of a Feature Pyramid Network object detector [12]. Several convolution layers are used to obtain feature maps of different sizes, including both high resolutions of low-level features and high semantic information of high-level features(©2017, IEEE).

CHAPTER 3 PROPOSED METHOD

In this chapter, there are three main sections. In Section 3.1, we describe the IOU tracker [13], as well as its shortcomings. In Section 3.2, we introduce the changes we made to IOU tracker to improve its tracking performance. In Section 3.3, we introduce the final multiple object tracking method that we proposed based on the changes of Section 3.2, and we show the algorithm in detail.

3.1 IOU tracker

The IOU tracker proposed by Bochinski et al. [13] depends on the Intersection Over Union (IOU) between each object bounding box in adjacent frames (see Figure 3.1). The input of the IOU tracker is the results of object detection, namely the bounding boxes (BBs) of the objects in the video. Then the bounding boxes in each frame are matched by IOU. For each frame, the method finds the detection bounding box which has the largest IOU with the bounding box of a currently tracked object, and then determines whether this IOU is greater than the threshold σ_l, and if so, add the bounding box to the trajectory of the tracked object. Otherwise, if a tracked object has no match, the method determines whether the maximum detection score in the track sequence is greater than the threshold σ_h, and whether the number of frames where the object appeared before the current frame is greater than the threshold t_{min}. If it is, it means that it is a normal tracking and that the object has left the scene, so the track is removed from the tracking queue and it is considered as a finished track. If not, the track is deleted. For the detection bounding boxes that could not be matched with the current tracks, they are considered as new objects, and their bounding boxes are added to the tracking queue as objects to be tracked in the next frames.

The IOU is defined as the area ratio of the intersection of two object BBs and the union of two object BBs as depicted in Figure 3.2. The threshold on the IOU is important as a slight change of threshold has a significant variation of the tracking result.

The advantage of the IOU tracker is that it is fast with a small computing cost. This algorithm can achieve tracking in a simple fashion with good performances. However, this method has some shortcomings that are worth more attention. Firstly, the IOU tracker much relies on the detector. If the detection results are not good, the tracking does not correspond to the expected result. Secondly, the IOU tracker cannot perform outstandingly when an occlusion exists, because the IOU will be low when the objects are being occluded. Thirdly,

tracker.png tracker.png tracker.png tracker.png

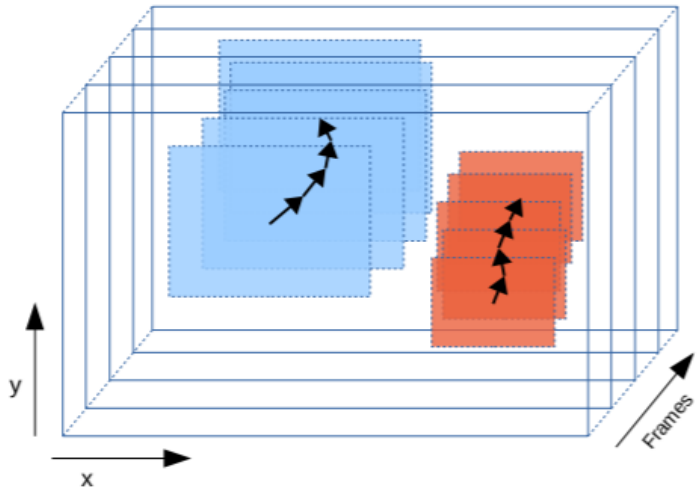


Figure 3.1 The data association of the IOU tracker [13](©2017 IEEE).

$$IOU = \frac{area\ of\ intersection}{area\ of\ union}$$

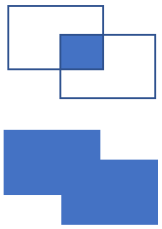


Figure 3.2 The definition of Intersection Over Union (IOU).

the IOU tracker does not consider the visual information of the object, which could improve the tracking through deep learning methods. When the objects in the video are in large number, ID switches will happen frequently due to the appearance feature missing in the IOU tracker.

3.2 Proposed changes to the IOU tracker

In order to improve the robustness and accuracy of the IOU tracker, we propose to add two components to the tracker. First, due to lack of prediction ability, the IOU tracker cannot perform well when the objects are occluded. Therefore, we modified the tracker by introducing a Kalman filter to predict the bounding boxes when occlusions appear. Second, appearance feature can be used to match the bounding boxes in the adjacent frames for

more robustness. From this perspective, we propose the use of re-id features as the visual information to match the bounding boxes by cosine similarity when appropriate.

3.2.1 Kalman filter

The principle of Kalman filter

The Kalman filter is an efficient autoregressive filter. It estimates the state of a dynamic system from combined information with many uncertainties over time. As long as it is a dynamic system with uncertain information, Kalman filtering can make educated guesses about what the system will do next. Even if there is noise information interference, Kalman filtering can usually figure out exactly what happened and find out the correlation between phenomenons that are not easy to detect. Another advantage of a Kalman filter is its small memory footprint (just keeps the previous state) and fast speed, which is ideal for real-time problems, like MOT, and for embedded systems.

Due to the above advantages, the Kalman filter can be applied to estimate the position and speed of the object in linear models with Gaussian noises. The Kalman filter uses the dynamic information of the target to remove the influence of noise to obtain a good position estimate of the target at the current frame (that is, the filtering process), and the future position estimate (that is, the prediction process).

Given the state of a moving object, including its position information and its speed information, the state of an object at time t can be represented by:

$$x_t = \begin{bmatrix} P_t \\ v_t \end{bmatrix}, \tag{3.1}$$

where P_t represents the position state of the object, and v_t is the speed state of the object. Considering the acceleration u_t and the adjacent time interval Δt, the state change of an object during the interval Δt can be written as the matrix

$$\begin{bmatrix} P_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{t-1} \\ v_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} u_t. \tag{3.2}$$

From equation 3.2, let F_t denote the state transition matrix, and B_t denotes the control matrix, respectively with

$$F_t = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, B_t = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}. \tag{3.3}$$

For a linear system, a simple estimate can be obtained based on the state at the previous time step. This is the state prediction formula of the Kalman filter and it is defined as

$$\hat{x}_t^- = F_t \hat{x}_{t-1} + B_t u_t. \tag{3.4}$$

The process of prediction will be affected by noise. The greater the noise, the greater the uncertainty of the predicted state. This change of uncertainty, Kalman filter defines it as the transmission of the noise covariance matrix, which represents the transmission relationship of the uncertainty at each time step. The transmission of the noise covariance matrix can be expressed as

$$\Sigma_t^- = F \Sigma_{t-1} F^T + Q, \tag{3.5}$$

where Σ denotes state error covariance and Q denotes the covariance matrix of the noise caused by the prediction model itself. The noise covariance matrix in the current state is inferred from the noise covariance matrix at the previous time step. In the process of transmission, it is also necessary to consider the influence of noise caused by the prediction model itself.

There is still a certain bias between the predicted value and the actual value. Since there is a bias, the predicted state needs to be corrected, which is the state update. When updating the predicted state, the bias between the predicted state and the observed value needs to be considered. If the observed value is Z_t, it can be expressed as

$$Z_t = H x_t + V, \tag{3.6}$$

where H denotes the measurement matrix, V denotes the measurement noise vector with a user-defined covariance R.

Using the error between the actual observation and the expected observation, the expected observation value can be corrected, and the correction is expressed as

$$\hat{x}_t = \hat{x}_t^- + K_t \cdot error = \hat{x}_t^- + K_t(Z_t - H\hat{x}_t^-) \tag{3.7}$$

with

$$K_t = \Sigma_t^- H^T (H \Sigma_t^- H^T + R)^-, \tag{3.8}$$

where K_t represents the Kalman gain. The function of the Kalman gain is to weigh the

covariance matrix Σ in the prediction state and the noise covariance matrix R introduced during the observation. If there is more confidence in the predictive model, then the corresponding covariance matrix will be smaller, and the Kalman gain K will be smaller, and if there is more confidence in the observation model, then the Kalman gain K will be larger.

According to the Kalman gain from Equation 3.8 and the transmission of the noise covariance matrix from Equation 3.5, Σ_t is updated with

$$\Sigma_t = (I - K_t H)\Sigma_t^-, \tag{3.9}$$

where I is the identity matrix. In the Kalman filter, Equation 3.4 and Equation 3.5 are used for prediction, and Equation 3.7, Equation 3.8 and Equation 3.9 are used for update.

Kalman filter in Multiple Object Tracking

The operation of the Kalman filter includes two stages: prediction and update. In the prediction stage, the filter uses the information of the previous state to make an estimate of the current state. In the update stage, the filter optimizes the predicted value obtained in the prediction stage by using the observed value of the current state to obtain a more accurate new estimate. In multiple object tracking, for the detection-based tracking framework, since the detection results are provided, they play the role of the observations. And then the Kalman filter predicts and updates the tracking state as explained in section 3.2.1.

The process of the Kalman filter in MOT is as follows: First, we initialize the current state of the object, and get the predicted state at the next time step according to the prediction equation (Equation 3.4, the acceleration is set to 0 in our case). And then, according to the noise in the prediction process, we obtain the noise covariance matrix at the current prediction state by Equation 3.5. Then comes to the update stage, by first obtaining a bias based on the current system observations and the state predicted at the previous time step by Equation 3.6. According to the transmission of the noise covariance matrix at the previous time step, a covariance estimate of the bias can be obtained. And then the Kalman gain is calculated based on the covariance matrix Σ in the prediction state and the noise covariance matrix R introduced during the observation by Equation 3.8. According to the Kalman gain and the error between the actual observations and expected observations, the current state value from the predicted state is updated and corrected by Equation 3.7, and this value can be used to predict the state at the next time step. Finally, the transmission noise is updated by Equation 3.9 through the Kalman gain and the estimated transmission of the noise covariance matrix.

3.2.2 Re-id feature

In the task of multiple object tracking, how to measure the similarity between objects in frames is a problem worth attention. There are many models aiming at solving this problem, including the modeling of motion, appearance, interaction, exclusion, and occlusion. The Kalman filter mentioned in Section 3.2.1 belongs to motion modeling, which considers the motion information from previous and current frames. In this section, we will discuss appearance modeling. There is no doubt that robust appearance features can improve the accuracy in measuring the similarity between objects, thus improving the tracking by reducing the number of identity switches.

An appearance model can be classified into two categories. One is the traditional method before deep learning, which extracts features through manually designed pixel operations. A representative method of this category is the KLT algorithm (Kanade-Lucas-Tomasi) [43] or color histograms. The other category is the CNN-based feature extraction based on deep learning where the pixel operations are learned. Based on the latter method, many Re-Id networks can be used as feature extractors.

Appearance models based on handcrafted features

The KLT algorithm is a feature point tracking algorithm based on the optical flow principle, which was first proposed by Lucas et al. [43], and then was improved by Shi et al. [14] later. It is a simple, real-time, and efficient tracking algorithm. There are three assumptions in the KLT algorithm: 1) the observed brightness of the objects is essentially constant, 2) the displacement of the objects between two nearby frames is small, 3) Neighboring pixels in the image move in a similar manner. The core idea of the KLT algorithm is to find some feature points suitable for tracking, and then perform the next step of tracking based on these feature points. The principle of this algorithm, in short, is to locate good features by checking the minimum eigenvalue of each 2 by 2 gradient matrix and use the Newton-Raphson method to track the features to minimize the difference between the two windows. Different from the optical flow method which directly compares the gray value of pixels, KLT compares a window around the pixels to find the most similar pixels. Figure 3.3 shows some features selected by the KLT algorithm.

Appearance models based on CNN-based feature extraction

CNN-based feature extraction achieves better results than manually designed algorithms. For multiple object tracking, a good method is to use a re-identification (Re-Id) network

Figure 3.3 The features selected by KLT [14](©1994 IEEE).

as a feature extractor. Re-Id networks extract features of objects from images and measure the distance of the features to determine whether they represent the same object or not. These features can be used to achieve object tracking. The features extracted from objects are represented by vectors and the distance of vectors can determine whether they represent the same object or not. If two detections are from the same object, the distance between the two vectors will be relatively small; if they are not from the same object, the distance between the two vectors will be relatively large. Re-id neural networks are trained using a loss specifically defined to obtain such result.

In the two nearby frames of a video, a number of bounding boxes of objects is obtained through the object detection method, and then features are extracted from the detection results through the Re-Id network. If the discriminatory ability of the Re-Id network is high enough, we could directly complete the pairing of all objects according to the distance between these vectors, even without any other tracking components and inference processes. However, it is only in an ideal situation. Although Re-Id can complete the function of object identification, its results are not enough perfect so that to replace other MOT algorithms. The Re-Id network has the ability to identify objects to improve the effectiveness of the existing algorithms in multiple object tracking. So more often, the Re-Id network is used as a component in MOT.

Re-Id feature network in our algorithm

In order to extract robust Re-Id features from objects, we used the Vehicle Feature Extractor [16] proposed by Wu et al. This feature extractor is based on a ResNet [44] trained on vehicle datasets, without the fully connection layer, by a Triplet loss, and the output of the last convolution layer is the feature we need.

The triplet loss is a loss function in deep learning used to train a network with inter-class samples with small differences. The triplet loss originated from FaceNet [15] that was proposed by Schroff et al. and it was used to learn a mapping for measuring the similarity of faces. The idea of the triplet loss is that the distance between intra-class samples is smaller and the distance between inter-class samples is larger. It is a loss function that is currently widely used. In FaceNet, the images of the face are directly mapped to the Euclidean space by an embedding method, and the method for optimizing this embedding can be summarized as constructing many groups of triplets, including Anchor, Positive and Negative, where Anchor and Positive have the same label and Anchor and Negative have different labels (in face recognition, that is, Anchor and Positive are the same individuals, but different from the Negative), through learning to optimize this embedding. The distance between Anchor and Positive in Euclidean space should be closer than the distance between Anchor and Negative. Figure 3.4 shows the learning with a triplet loss. The equation of the triplet loss L is given by

$$L = \sum_{i}^{N} [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+, \tag{3.10}$$

where x_i^a denotes the Anchor, x_i^p denotes the positive sample, x_i^n denotes the negative sample and α denotes the margin between positive and negative pairs, + denotes the hinge loss.

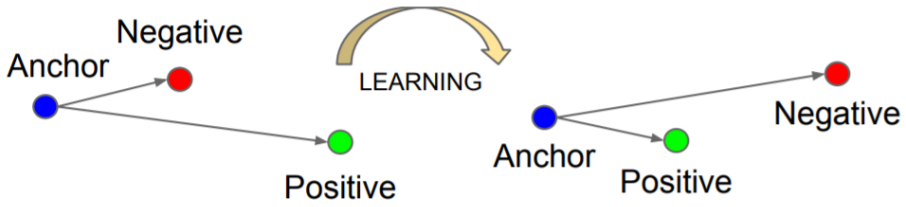


Figure 3.4 The triplet loss minimizes the distance between the Anchor and the Positive and maximizes the distance between the Anchor and the Negative [15](©2015 IEEE).

The triplet loss can usually obtain better features (more discriminative) than a classification loss. Another advantage is that a margin threshold can be manually set and changed (α) in the triplet loss. This margin threshold can control the distance between positive and negative

samples. After the feature is normalized, it is more convenient to determine whether they are the same label through the threshold. The disadvantage of the triplet loss is that it converges slowly and is more likely to overfit than classification.

In the Vehicle Feature Extractor [16], an adaptive feature learning technique has been proposed to collect positive and negative samples from unlabeled videos. Since one object cannot appear at more than one location in the same frame, any objects from the same frame can be sampled as negative samples. And the objects are moving over time, thus the same object in different frames can be identified with the IOU and considered as a positive sample. Figure 3.5 shows the positive pairs and negative pairs in the adaptive feature learning technique. The input of the Vehicle Feature Extractor is the object cropped from the video and the output is a vector with 2048 elements representing the appearance features of the object. In our algorithm, we crop the objects using the detection bounding box from each frame of the video.

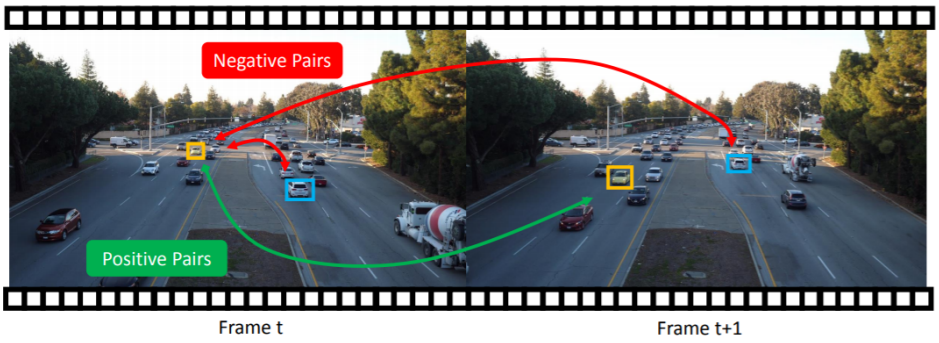


Figure 3.5 The positive pairs and negative pairs in the Adaptive Feature Learning technique [16](©2018 IEEE).

3.3 Our proposed multiple object tracking method

Based on the IOU tracker, we propose to add a Kalman filter component for prediction and a CNN-based feature extractor for data association. Before solving the multiple object tracking task, there are some assumptions that need to be clarified. First, the motion of the object should be at a approximately constant speed. Secondly, the same object will not be far away in two adjacent frames, that is the speed of the objects should not be too fast. In addition, the appearance of the objects should be approximately constant in the short term.

In the tracking task, when we obtain the detections from the detector, we use the IOU to achieve the preliminary matching of the bounding boxes. If the detection result is excellent

and the bounding boxes in the frame are far from each other, matching based on IOU will be good as expected. However, this matching algorithm that relies on the IOU just uses the information of the position and size of the bounding box. If the objects in the video are in large numbers, there will be overlapping of the bounding boxes, thus resulting in potential identity switches.

In order to reduce the number of identity switches, we add a prediction component after the IOU matching with the use of the Kalman filter. The Kalman filter is applied to unsuccessful matches. The motion estimation obtained in the prediction stage of the Kalman filter is associated with the observation (detection), and the cost matrix is the IOU between the predicted position of the object in the current frame and the bounding box in the current frame. If the match is successful, the Kalman filter is updated. Tracking trajectories that fail to match are regarded as lost, and detections that fail to match are regarded as new trajectories.

The Kalman filter can solve the problem of the occlusion of an object in the short term. Because when the object is occluded, the occluder is detected instead of the original object, but the Kalman filter can predict the position of the object based on its previous trajectory information. Then when the occlusion finishes, the predicted bounding box should have a large value of IOU with the object, and the correct association can be restored.

The difference between our method and SORT is that the assignment in our algorithm is based on the IOU of the bounding box using a greedy approach while the assignment in SORT relies on the Hungarian algorithm [18]. The reason why we do not use the Hungarian assignment algorithm is it achieves as many matches as possible instead of matching as accurately as possible. We believe that in order to avoid identity switch, we should assign the bounding boxes as accurately as possible by considering first the most confident matches.

The Kalman filter is useful if the object has been occluded partially and temporarily, since it is based on a motion model. The prediction of the bounding box is based on the tracking history and motion information. If the distance between the same object is large in the adjacent frames or if the occlusion is full for several frames, the Kalman filter cannot work as well as expected. In order to handle this problem, we propose to use a CNN-based Re-Id network. The Re-Id network is based on the appearance features of the object, and then measures the distance of the features to determine whether they represent the same object. If the Kalman filter cannot predict the trajectory well, we use the Re-Id component, where we extract the features of the unmatched objects, as well as the features of the objects in the last frame of the trajectory, and then we compute the similarity of these features. If the similarity is high, the unmatched object will be added into the trajectory, leading to less identity switches.

Because we extract the features of all unmatched objects, the Re-Id network component is not limited by spatial distance. It means that although the object moves significantly over a few frames, this component can achieve the data association successfully because it is purely appearance-based.

The measurement of feature distance that we used in our algorithm is the cosine similarity, which evaluates the similarity by calculating the cosine of the angle between two vectors. Figure 3.6 shows the cosine similarity of two vectors in two dimensions. The range of the value of the cosine similarity is $[-1, 1]$. The larger the value, the more similar the two vectors. The equation of the cosine similarity between vector A and B is

$$cos\theta = \frac{A \cdot B}{\|A\| * \|B\|}. \tag{3.11}$$

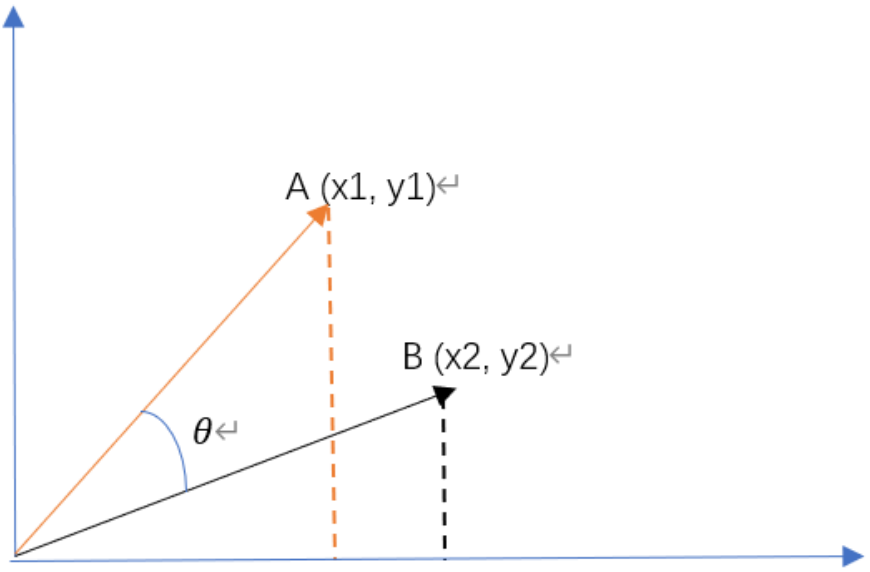


Figure 3.6 The cosine similarity of two vectors in two dimensions

The whole process of multiple object tracking in our algorithm is the following:

- First, we compute the intersection over union (IOU) of the bounding boxes in adjacent frames. If the IOU of a bounding box B_i^t in the current frame t and a track T_j^{t-1} is higher than the threshold s_l, B_i^t will be added to the matching track T_j^{t-1}.
- Secondly, if the IOU of a bounding box B_i^t and a track T_j^{t-1} is lower than the threshold s_l but higher than the threshold s_k, and if the length of T_j^{t-1} is longer than 5 frames,

we will apply the Kalman filter to predict the bounding box of T_j^{t-1} in the frame t, that is PB_j^t. If the IOU of PB_j^t with B_i^t is larger than the threshold s_{pk}, we will add it to the track T_j^{t-1} and it will replace the original bounding box in the current frame. If the IOU of PB_j^t and B_i^t is lower than the threshold s_k or the bounding box fails to be matched through the Kalman filter, it will be added to the unmatched object set.

- Thirdly, if the IOU and the Kalman filter cannot work well, we start to achieve data association through appearance features. We obtain the appearance features by the Vehicle Feature Extractor [16], where the neural network is trained by the triplet loss and the features of each object are recorded using a vector with 2048 elements. Through the Vehicle Feature Extractor [16], we obtain the features of objects in each track, as well as for the unmatched object set. In order to match the last bounding box in the track T_j^{t-T} (where $t-T$ is the time of the last observation) and a bounding box B_i^t in the unmatched object set, we compute the similarity of these features by cosine similarity. If the similarity is high, B_i^t will be added to the track T_j^{t-T}. Otherwise, we assume that track T_j^{t-T} is finished. If the maximum of the confidence score of the bounding boxes in the track is higher than S_h and the length of the track is more than t_{min}, we will consider this tracking sequence as a successful tracking.

  The bounding box in the unmatched object set B_i^t will start a new track.

The detailed description of our method is presented in Algorithm 1. Where B^t denotes a set of detection results at frame t, B^i denotes the i_{th} detection at that frame, T_j denotes an active tracking sequence.

Algorithm 1 Framework of tracking algorithm proposed

Input:

The set of the detection at frame t, $B^t = \{B^0, B^1, ..., B^t\} = \{\{B_0^0, B_1^0, ..., B_{N-1}^0\}, \{B_0^1, B_1^1, ..., B_{N-1}^1\}, ...\}$;
s_l, s_h, s_k, s_{pk}, t_{min}.

for $t = 0$ to t **do**
  for $T_j^{t-1} \in T^{t-1}$ **do**
    $d_{best} = d_j$, where $\max(IOU(B_i^t, T_j^{t-1})), B_i^t \in B^t$
    $prediction = kf[f]$
    if $IOU(d_{best}, T_j^{t-1}) > s_l$ **then**
      add d_{best} to T_j^{t-1}
      remove d_{best} from B^t
    else
      if $IOU(d_{best}, T_j^{t-1}) > s_k$ **then**
        $PB_j^t = kf[j]$
        if $IOU(PB_j^t, d_{best}) > s_{pk}$ **then**
          add PB_j^t to T_j^{t-1}
          $time+ = 1$
        else
          add d_{best} to unmatched object set
        end if
      else
        $vis[i]$ = visual tracker
        $time+ = 1$
        add $vis[i]$ to T_j^{t-1}
      end if
    end if
  end for
  for $T_j^{t-1} \in T^{t-1}$ **do**
    if $highest - score(T_j^{t-1}) > s_h$ AND $len(T_j^{t-1}) > t_{min}$ **then**
      add T_j^{t-1} to T_f
      remove T_j^{t-1} from T_a
    end if
  end for
  for $B_i^t \in B^t$ **do**
    start new track with B_i^t and insert into T_a
  end for
end for
return T_f;

CHAPTER 4 GENERAL DISCUSSION

In this chapter, we will present the results of our experiments. There are five main sections. In Section 4.1, we describe the dataset that we used, that is UA-DETRAC [1]. In Section 4.2, we describe the evaluation metrics in our method. In Section 4.3, we introduce the detail of the implementation for the experiments. In Section 4.4, we present the tracking results obtained by our method and compare the results with other state-of-the-art methods on the dataset. And we have some discussion about the experimental results in this section. In Section 4.5, we analyze the results and detail the contribution of each part of our method.

4.1 Datasets

The UA-DETRAC dataset was proposed by Wen et al [1]. The whole dataset includes 100 challenging video of real-world traffic scenes, which splits into 60 training and 40 test videos. We tested our method on this dataset which consists of a total of 40 videos of traffic scenes. The dataset includes videos that were captured with a fixed camera for different traffic scenes and conditions, including urban highway, traffic crossings, and T-junctions. The videos in the datasets are recorded at 25 frames per seconds (fps) and the resolution of the images is $960 * 540$ pixels. The figure 4.1 shows sample frames of the UA-DETRAC dataset. There are three levels of difficulties among the 40 test videos, namely easy (10 videos), medium (20 videos), and hard (10 videos).

Compared with other existing tracking datasets, the UA-DETRAC dataset is more diverse, because the data was collected at various locations with several different illumination conditions and filming angles. There are four types of illumination conditions, namely sunny, rainy, cloudy and night. The category of vehicles is also varied, including car, bus, van, and others (such as trucks and road tankers).

4.2 Evaluation metrics

How to measure the tracking performance needs to include following elements

- All targets that appear should be found across frames;
- The target position should be as consistent as possible with the real accurate target position;

Figure 4.1 Sample frames in the UA-DETRAC dataset [1]. The left one shows a daytime scenario and the right shows at nighttime scenario.

- Each target should be assigned a unique identity (ID), and the ID assigned to the target should remain the same throughout a video sequence.

In our experiments, we used the CLEAR MOT metrics and the PR-MOT metrics. MOTA (multiple object tracking accuracy) is one of the main evaluation metric in MOT Challenge [45], and it reflects the number of correctly detected targets and the accuracy of the assigned target identities, and it is used to count the accumulation of errors in tracking, including False Positive (FP), False Negative (FN), Identity Switch (IDSW). MOTA is expressed as

$$MOTA = 1 - \frac{\Sigma_t(FN_t + FP_t + IDSW_t)}{\Sigma_t GT_t} \tag{4.1}$$

where t is the frame index, GT_t is the number of ground truth objects, FN_t is the number of false negative and FP_t is the number of false positive in the t^{th} frame, $IDSW_t$ is the number identity switches between frames. The range of MOTA is $(-\infty, 1)$, therefore the MOTA may be negative. Indeed, when the errors generated by the tracker exceed the number of objects in the scene, the MOTA will be negative.

MOTP(multiple object tracking precision) is another evaluation metric mentioned in MOT Challenge [45], and it represents the accuracy of target location. Therefore, it depends on the location accuracy of the detector. The MOTP value is the ratio of the position estimation error of the matched target-prediction pair in all frames and the number of matches. It is expressed as

$$MOTP = \frac{\Sigma_{t,i} d_{t,i}}{\Sigma_t C_t} \tag{4.2}$$

where $d_{t,i}$ represents the distance between target i and its ground truth position in the frame t, and C_t denotes the the number of matches in the t^{th} frame.

The MOTA metric is affected by the number of false positives, false negatives, as well as by the identity switches with equal weights. It means that the impact of identity switch is not obvious in the MOTA metric. Furthermore, it is not convincing to compare the performance of tracking based on the different settings of detections with different FN and FP values. An excellent tracker should achieve excellent performance with any setting of for the detection threshold. Therefore, we use also the tracking metrics of PR-MOT [1], the average MOTA and MOTP scores over the PR curve of the detector. The figure 4.2 shows the PR-MOTA curve. The PR-MOTA score can be computed by

$$\text{PR-MOTA} = \frac{1}{2}\int_C \Psi(p,r)ds \tag{4.3}$$

where $\int_C \Psi(p,r)$ represents the MOTA score of the tracking algorithm determined by the precision-recall (PR) curve of detection.

The score of the PR-MOTP is computed similarly to that of the PR-MOTA.

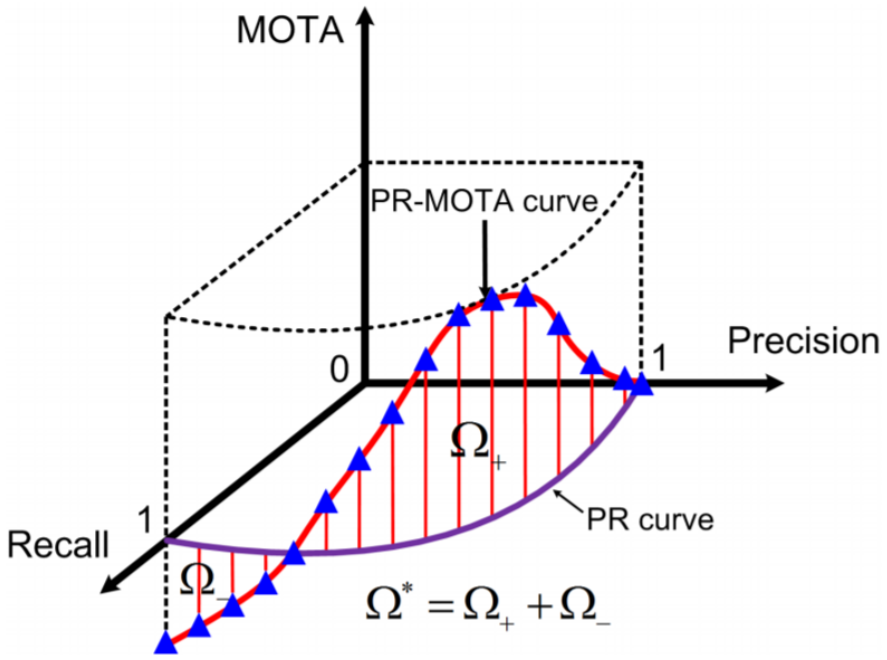


Figure 4.2 The red curve is the PR-MOTA, the purple curve is the precision-recall curve representing the performance of the detector, and the blue triangles are sampling points (MOTA tracking results) on the PR-MOTA curve. [1].

The PR-MOT metrics represent the performance of trackers better since they consider the detector under several threshold settings. The PR-MOT metrics pay more attention to tracking performance while MOT metrics can be affected by the settings of detectors easily.

Therefore, in the experiment results, we use the PR-MOT metrics as the main measurement and then the MOT metrics, which are the max over the PR curve.

4.3 Implementation details and parameters

The implementation was done completely in Python. We used the IOU tracker [13] as our backbone because it shows very good tracking performance while being very fast. In addition to the IOU tracker backbone, the whole method includes the Kalman filter component and the Re-Id feature component. For our experiments, for the Re-Id features, we used the Vehicle Feature Extractor [16] pretrained on multiple datasets of vehicles by triplet loss.

In our experiment, there are various parameters that should be tuned. we used the training set to find good parameters. First, the low detection threshold is the parameter that filters the detection results by the confidence score. In order to compute the PR-MOT metrics, we should set this threshold variable from 0 to 1.0 by step of 0.1. And then, the high detection threshold s_h, is the parameter that filters the trajectories according to the maximum of the confidence score over a whole trajectory. In our experiments, we tried 0.95 and 0.9, and the tracking performance is better if the threshold is 0.9. If the high detection threshold is too high, there will be many successful trajectories filtered and the false negative will increase. If the high detection threshold is too low, there will be many unsuccessful trajectories kept and the false positive will increase. The intersection-over-union threshold s_l, is the parameter that matches the bounding boxes in the adjacent frames by the value of intersection-over-union. After referring to the parameter from [2], we tried various values and the best value found was 0.4. If this threshold is too high, the bounding boxes are rarely matched by IOU, and then it increase the computation and the responsibility of the Re-Id component.

The kalman-iou threshold s_k, is the parameter that selects bounding boxes in the adjacent frames by the value of intersection-over-union. The selected bounding boxes are used in the Kalman filter to predict the position of new bounding boxes. This parameter should be close to the intersection-over-union threshold. Bounding boxes whose IOU values are between these two parameters are selected to be used in the Kalman filter component. We set the kalman-iou threshold s_k as 0.3 in our experiment. The prediction-kalman-iou threshold s_{pk}, is the parameter for matching the predicted bounding boxes with the original ones by the value of intersection-over-union. The predicted bounding boxes will replace the original ones if the value of IOU is larger than this threshold. We set this parameter as 0.9 in our experiment. The cosine similarity threshold is the parameter that matches the trajectory with unmatched bounding boxes using Re-Id features by the value of cosine similarity. We tried 0.95 and 0.9 in the experiment, and the tracker performs better if the parameter is set as 0.9. If the

cosine similarity threshold is too high, there will be fewer matches by the Re-Id component. And if the parameter is too low, incorrect matches will happen frequently. The minimum track length threshold t_{min}, is the parameter that specify the minimum length requirement to determine whether a tracking is successful. Considering the length of objects existing in the video and referring to the parameter in [2], we set it as 12 in our experiment. That means only tracks with length of more than 12 frames will be considered as successful tracking. Table 4.1 give the parameter values used in the experiment.

s_h	s_l	s_k	s_{pk}	cosine similarity	t_{min}
0.9	0.4	0.3	0.9	0.9	12

Table 4.1 Best parameters for our tracker for the UA-DETRAC dataset.

4.4 Results and discussion

We have compared our method with recent state-of-the-art methods on the UA-DETRAC dataset. The experimental results are given in Table 4.2. We report the PR-MOTA and PR-MOTP score and the best MOTA and MOTP score among all the detection thresholds for each detector. We tested our tracking method on two detection methods, that is SpotNet [41] and Faster R-CNN [10].

We outperform our baseline, IOU-tracker, by a very significant margin and we are close to the PR-MOTA score of the state-of-art and getting state-of-art results on the dataset with the PR-MOTP metric. We report an improvement in the value of 10.8% in PR-MOTA and 13.8% in PR-MOTP over the IOU-tracker baseline. The PR-MOTA values obtained by our method based on SpotNet is 30.6%, which is much better than the baseline tracker using the same detection, and this result is very close to the state-of-art result of 30.7% obtained by the VIOU tracker based on Mask R-CNN. The PR-MOTP values obtained by our method based on SpotNet is 42.7%, which much better than the baseline tracker using the same detection, and this result outperforms the state-of-art result obtained by the VIOU tracker based on Mask R-CNN. The Kalman filter and Re-Id feature components in our method have proven to be effective to improve the performance of the baseline IOU-tracker. Figure 4.3 shows the PR curve of the IOU tracker and our method on the SpotNet detection. It shows that our proposed method reaches a higher recall as well as a higher precision. The PR curve is based on various thresholds from 0 to 1, so the maximum of recall in the baseline may not be high enough. Figure 4.5 gives the resulting PR-MOTA curve and Figure 4.4 describes the resulting PR-MOTP curve on the baseline and our method. The area under the PR-MOT

curve describes the value of PR-MOT. These figures show that the MOTP is stable with precision and recall, because it is only based on the performance of the detector. They show that the MOTA in our method is not good as expected when the recall is very high.

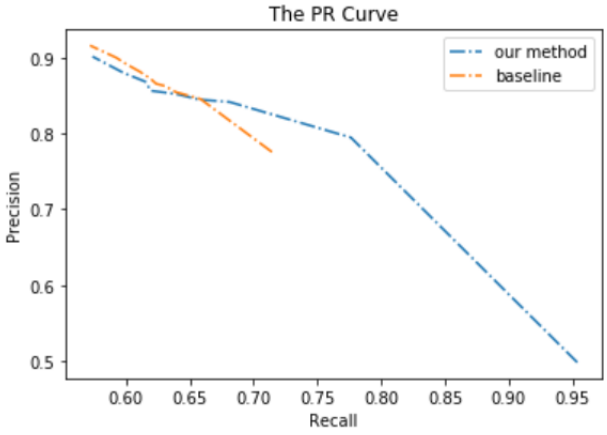


Figure 4.3 The figure shows the PR curve of the baseline and our method.

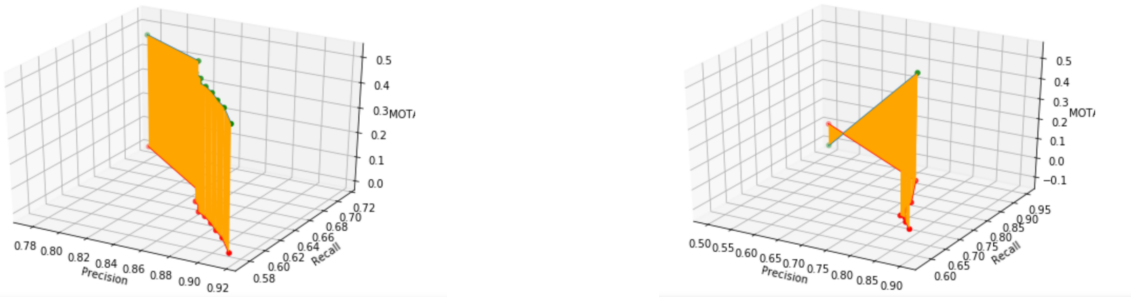


Figure 4.4 The left figure shows the 3D PR-MOTA curve of the baseline and the right one shows the PR-MOTA curve of our method.

However, in terms of the MOTA metric, our method only improves the performance marginally, increasing the MOTA score to 53.5% from 51.6% on the baseline tracker. And the MOTP score decreases slightly from 75.6% on the baseline tracker to 73.8%. And the MOTA and the MOTP value obtained by our method cannot reach the state-of-art results on the datasets. We think that the detection methods influence this value significantly. However, our proposed tracker performs more consistently for various detection thresholds.

In order to describe the tracking ability of our method, we tested our method on the three subsets of the UA-DETRAC dataset, easy, medium, and hard subsets. And we compared the PR-MOTA and the PR-MOTP score obtained by our method with state-of-the-art methods mentioned in [1].

Figure 4.5 The left figure shows the 3D PR-MOTP curve of the baseline and the right one shows the PR-MOTP curve of our method.

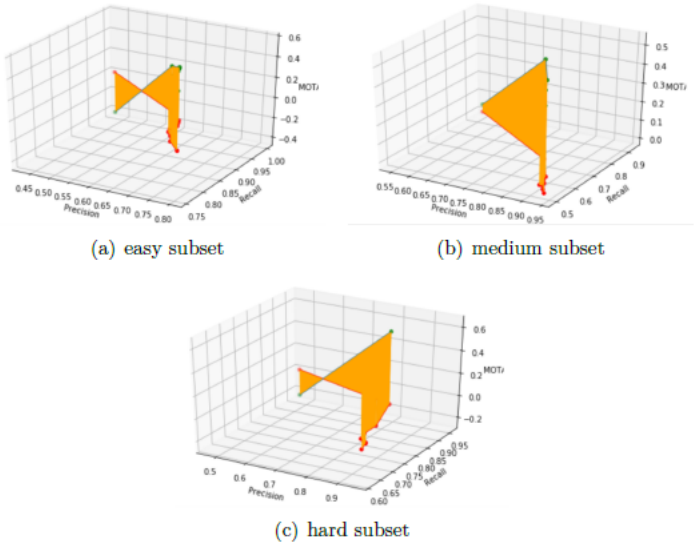
Method	Detector	PR-MOTA	MOTA	PR-MOTP	MOTP
GMPHD-KCF#	CompACT	14.8%	-	36.0%	-
GOG#	CompACT	14.2%	44.4%	37.0%	80.8%
DCT#	RCNN	11.7%	38.4%	38.0%	-
MHT#	Faster R-RCNN	14.5%	58.2%	32.5%	78.4%
KIOU#	EB	21.1%	**62.1**%	28.6%	**81.9**%
IOU*	RCNN	16.0%	-	38.3%	-
VIOU*	Mask R-CNN	**30.7**%	-	37.0%	-
IOU	SpotNet	19.8%	51.6%	28.9%	75.6%
our method	SpotNet	30.6%	53.5%	**42.7**%	73.8%

Table 4.2 result on the overall UADETRAC-Test dataset. The results of # were taken from [1]. The results of * were taken from [2]

		PR-MOTA			PR-MOTP		
Method	Detector	easy	medium	hard	easy	medium	hard
GOG#	CompACT	25%	14.1%	5.5%	47.0%	35.1%	32.8%
DCT#	RCNN	23.0%	11.8%	4.7%	46.3%	35.7%	35.7%
MHT#	Faster R-CNN	23.0%	14.2%	9.5%	24.1%	31.8%	29.0%
GOG#	ACF	22.9%	10.7%	1.7%	**48.6**%	35.1%	32.8%
KIOU#	EB	**36.4**%	19.9%	13.1%	37.6%	26.3%	25.6%
IOU#	EB	34.0%	18.2%	11.9%	37.8%	26.6%	26.2%
our method	SpotNet	30.4%	**21.2**%	**18.8**%	36.4%	**45.4**%	**46.1**%

Table 4.3 PR-MOTA score on easy, medium and hard subsets of the UA-DETRAC dataset. The results of # were taken from [1].

From Table 4.3, In terms of the PR-MOTA score, we can find that the KIOU method based on EB detection performs best on the easy subset, but our method performs much better in the medium and hard subsets than other state-of-the-art methods. The PR-MOTA score obtained by our method is 21.2% in the medium subsets and 18.8% in the hard subsets. Figure 4.6 shows the 3D PR-MOTA curve of our method on the easy, medium and hard subsets. In terms of the PR-MOTP score, we can note that the GOG method based on ACF detection performs much better than other methods on the easy subset, but our method performs best in the medium and hard subsets. The PR-MOTP score obtained by our method is 45.4% in the medium subsets and 46.1% in the hard subsets. Figure 4.7 shows the 3D PR-MOTP curve of our method on the easy, medium and hard subsets. The reason why the PR-MOTP score is better in medium and hard subsets is that the MOTP for the subsets are based on the matched objects, so even if fewer objects are matched, the ones that are matched can be better located on average. Therefore, SpotNet is better at localizing more precisely objects in difficult scenarios. The figures of PR-MOTA and PR-MOTP for the subsets are like these for the overall dataset. The MOTP is stable with precision and recall, and the MOTA is not good as expected when the recall is very high.

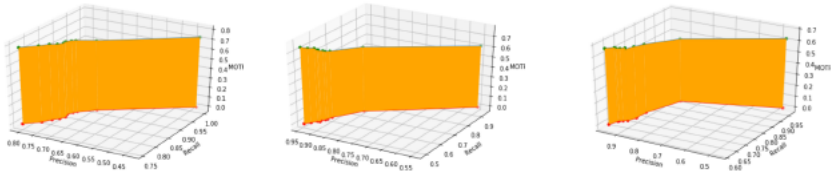


(a) easy subset (b) medium subset

(c) hard subset

Figure 4.6 The 3D PR-MOTA curve of the easy subset, medium subset and hard subset.

In our experiments, we used SpotNet [41] as our detector, which ranks first on the UA-DETRAC detection dataset. There are two issues with the detection results of SpotNet. Firstly, the confidence score of the detection results of SpotNet is low. Second, detection results of SpotNet record the top 100 objects in each frame. We tried to implement our tracker on the original results of SpotNet, but the performance of tracking is not very good



(a) easy subset (b) medium subset (c) hard subset

Figure 4.7 The 3D PR-MOTP curve of the easy subset, medium subset and hard subset.

because of the way the PR-MOT metrics works (varying the detection threshold from 0 to 1). The number of objects in each frame is less than 100 and if we use the original detection, there are many false positive in the tracking results when the detection threshold is low, which affects the MOTA score and the PR-MOTA score. Considering that the detection results for Faster R-CNN record only the top 50 objects, to have a fair comparison, we ranked the objects in each frame by confidence score and selected the top 50 objects as our detection results with SpotNet. As for the problem of low confidence score, we multiplied by 3 the original scores. If we use the original score, when we calculate the PR-MOTA score where we set different thresholds for the detection score, there are very few trajectories with a high threshold, resulting in several successful trajectories being filtered. Furthermore, for high detection threshold the recall of SpotNet is low so many objects are missing. In sum the original scores of SpotNet are not compatible with the PR-MOT evaluation metrics, because they tend to be low. Rescaling was necessary.

Figure 4.8 shows the tracking result of 125^{th} frame and 185^{th} frame in MVI 40701. We can note that the tracking for the objects with id 92 and 83 performs well, although they move a long distance and are occluded by each other in the video. However, the tracking for the objects with id 17 and 27 is not as good. We think those failures can be explained by the appearance of the object with id 27 and 17 is occluded by other objects and close to other objects.

4.5 Ablation study

In order to detail the contribution of each component in our model, we performed an ablation study on the UA-DETRAC dataset. For this study, we chose SpotNet as the detector and explore the Kalman filter component and Re-Id feature component contributions by their tracking performance. Table 4.4 shows that even though the Kalman filter helps, the biggest contribution comes from combining the Re-Id feature with it. We can note that the MOTA

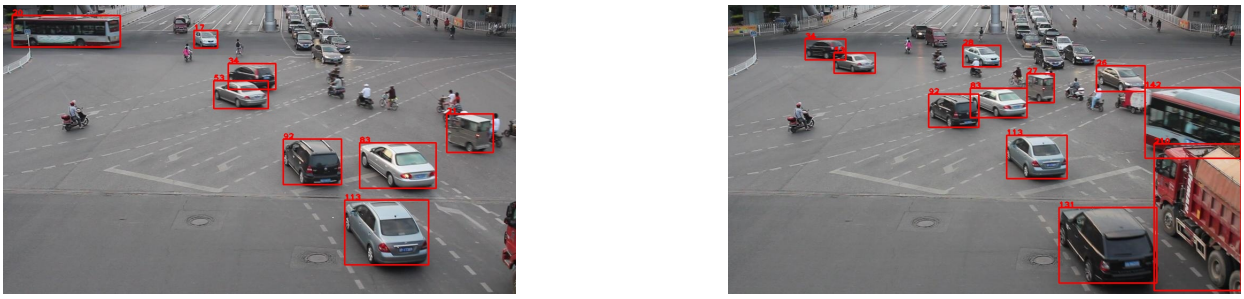


Figure 4.8 These two figures show the tracking result of 125^{th} frame and 185^{th} frame in MVI 40701. Although the gap of the frames is 60, the tracking performance keeps well.

score has been slightly increased and the MOTP score has decreased, but the PR-MOTA score and the PR-MOTP score is improved significantly.

Kalman filter	Re-Id feature	PR-MOTA	MOTA	PR-MOTP	MOTP	ID switches
		19.8%	51.6%	28.9%	75.6%	5104
√		23.1%	52.5%	36.2%	73.6%	3882
√	√	30.6%	53.5%	42.7%	73.8%	3634

Table 4.4 Ablation study on the overall UADETRAC-Test dataset

Re-Id feature component When we explore the detail of the tracking results, we note that when the Re-Id feature component is added to the model, the tracking performance is not very good if the detection threshold is low. The reason we think explains this is that if the detection threshold is low, there will be many detections with a low confidence score and perhaps these detections do not exist in the ground truth (not actual objects). And the Re-Id feature component may match these detections and increase the false positive significantly. These incorrect matches happen frequently when the low detection threshold is 0. However, if the low detection threshold is high enough, the Re-Id feature component can improve the tracking performance, and the PR-MOTA score and the PR-MOTP score can increase as much as by 7.5% and 6.5% separately, compared to our tracker without it. The MOTA score is improved in each threshold except 0. We found that the model with the Re-Id feature component can decrease false negative, identity switch and increase true positive. Our PR-MOTA could probably be better than the state-of-the-art if Re-Id could be disabled in some way when the number of detections in frames is too large.

Kalman filter component When we add the Kalman filter to the baseline model, we find that the Kalman filter can reduce identity switch significantly. The number of identity switches presented in Table 4.4 is the minimum among all the detection thresholds. Compared to our tracker without the Kalman filter component, the identity switches can decrease by

around 24%. The reason we think that explains this, is that the Kalman filter can connect the unmatched objects with aborted trajectories and thus avoid starting a new trajectory. The Kalman filter can predict the position of the object by the previous trajectory, which improves the precision of the bounding box.

CHAPTER 5 CONCLUSION

In this chapter, we will present the conclusions of our work. In Section 5.1, we describe the summary of our contribution. In the Section 5.2, we discuss the limitations of our model, and some ideas to deal with them. Finally, in the Section 5.3, we present possible future research based on our work.

5.1 Summary of contributions

In this work, we presented a new multiple object tracking method based on the IOU tracker. In order to improve the tracking performance, we added a Kalman filter component and a Re-Id feature component into the IOU tracker. The Kalman filter can predict the position of the object if the object is occluded, and it can match a trajectory with unmatched objects and thus can reduce identity switch significantly. We have conducted an ablation study and confirmed the contribution of the Kalman filter component to the performance of our tracker in Section 4.5. The Re-Id feature component allows matching a trajectory with unmatched objects by using an appearance feature. It reduces the identity switch and increases the true positive, namely the number of correct matches. We use the cosine similarity as the matching criterion for the Re-Id feature. In the ablation study, we note that the Re-Id feature component improves the tracking performance significantly. We tested our method on the UA-DETRAC dataset and the final tracking results show that our method could reach performance close to the recent state-of-the-art methods, while at the same time outperforming significantly the base IOU tracker.

5.2 Limitations

Our method also comes with new challenges, although we have improved the tracking performance of the IOU tracker. In the Re-Id feature component, the computation cost of the appearance feature is expensive. The size of the Re-Id feature is 2048 for each object, and thus it takes a long time to compute and also much memory space to record the Re-ID features. The Re-Id feature component cannot work if the detection threshold is 0. In that case, the Re-Id feature component can result in many false positives.

5.3 Future Research

As we mentioned in the limitations, due to the expensive computation cost and large space of the Re-Id features, we could research a way to decrease the computation cost and space. In our model, we did not consider the physical distance of the objects, which could be a prerequisite for the Re-Id feature matching. It means that before computing the Re-Id feature of the objects and trajectories, we should consider the physical distance in pixels in the image. If the physical distance is too large, we could consider that they are not the same object and not compute the feature.

REFERENCES

[1] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *arXiv preprint arXiv:1511.04136*, 2015.

[2] E. Bochinski, T. Senst, and T. Sikora, "Extending iou based multi-object tracking by visual information," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[3] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[4] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," *arXiv preprint arXiv:1409.7618*, 2014.

[5] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[6] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.

[7] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 941–951.

[8] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3988–3998.

[9] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[11] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[13] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[14] J. Shi *et al.*, "Good features to track," in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, 1994, pp. 593–600.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[16] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu, and S.-Y. Chien, "Vehicle re-identification with the space-time prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 121–128.

[17] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[18] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[19] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[20] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[21] H. Kieritz, W. Hubner, and M. Arens, "Joint detection and online multi-object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1459–1467.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[23] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 366–382.

[24] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4705–4713.

[25] Z. He, J. Li, D. Liu, H. He, and D. Barber, "Tracking by animation: Unsupervised learning of multi-object attentive trackers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1318–1327.

[26] Y.-c. Yoon, A. Boragule, Y.-m. Song, K. Yoon, and M. Jeon, "Online multi-object tracking with historical appearance matching and scene adaptive detection filtering," in *2018 15th IEEE International conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[27] Y.-C. Yoon, D. Y. Kim, K. Yoon, Y.-m. Song, and M. Jeon, "Online multiple pedestrian tracking using deep temporal appearance matching association," *arXiv preprint arXiv:1907.00831*, 2019.

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

[29] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[30] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2013.

[31] J. Zhang, S. Zhou, X. Chang, F. Wan, J. Wang, Y. Wu, and D. Huang, "Multiple object tracking by flowing and fusing," *arXiv preprint arXiv:2001.11180*, 2020.

[32] G. Brasó and L. Leal-Taixé, "Learning a neural solver for multiple object tracking," *arXiv preprint arXiv:1912.07515*, 2019.

[33] J. Zhang, S. Zhou, J. Wang, and D. Huang, "Frame-wise motion and appearance for real-time multiple object tracking," *arXiv preprint arXiv:1905.02292*, 2019.

[34] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3588–3597.

[35] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942–7951.

[36] S. Lee and E. Kim, "Multiple object tracking via feature pyramid siamese networks," *IEEE Access*, vol. 7, pp. 8181–8194, 2018.

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[39] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[40] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6569–6578.

[41] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Héritier, "Spotnet: Self-attention multi-task network for object detection," *arXiv preprint arXiv:2002.05540*, 2020.

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

[43] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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