

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**MULTIPLE PEDESTRIAN TRACKING BY KERNELIZED
CORRELATION FILTERS**

Master's Thesis

IBRAHİM ABU ARAFEH

İSTANBUL, 2019

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING**

**MULTIPLE PEDESTRIAN TRACKING BY
KERNELIZED CORRELATION FILTERS**

Master's Thesis

IBRAHİM ABU ARAFEH

Supervisor: PROF. DR. NAFİZ ARICA

İSTANBUL, 2019

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**The Graduate School of Natural and Applied Sciences
Computer Engineering**

Title of the Master's Thesis : Multiple Pedestrian Tracking by Kernelized Correlation Filters
Name/Last Name of the Student : Ibrahim ABU ARAFEH
Date of Thesis Defense : 11 January , 2019

The thesis has been approved by The Graduate School of Natural and Applied Sciences.

Assist. Prof. Dr. Yücel Batu Salman
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Tarkan Aydın
Program Coordinator
Signature

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members:

Signature

Thesis Supervisor
Prof. Dr. Nafiz Arica

.....

Member
Assist. Prof. Dr. Övgü Öztürk

.....

Member
Assist. Prof. Dr. Mert Bal

.....

ACKNOWLEDGEMENTS

I would like to thank my mother and father for all their patience and support until this moment.

I would also like to thanks my supervisor Dr. Nafiz Arica for his guidance through out this research.

İstanbul, 2019

Ibrahim ABU ARAFEH



ABSTRACT

MULTIPLE PEDESTRIAN TRACKING BY KERNELIZED CORRELATION FILTERS

Ibrahim ABU ARAFEH

Computer Engineering
Supervisor: Prof. Dr. Nafiz ARICA

January 2019, 57 Page

In this thesis, we study multiple pedestrian tracking by detection using Kernelized Correlation Filter (KCF) based object tracking method. In order to improve the data-association results to object detections, Intersection Over Union (IOU) method is utilized along with Hungarian algorithm. The method performance is evaluated on the Multi-Object Tracking (MOT) benchmark, showing that combining both methods leads to a better accuracy of object tracks on the sake of increased computation complexity. Among other methods, this thesis also applies Structural Similarity Index (SSIM) to be combined with IOU for data association in some scenarios. In night scenarios, this model which utilizes the luminance changes in images manages to give better results than the other adopted model (IOU) which gives better overall results among the other video sequences. This work shows that adopting a robust appearance model can perform better in multiple object tracking than a motion model that utilizes the same data association method in video data containing multiple pedestrians.

Keywords: Multiple Object Tracking, Kernelized Correlation Filter, Computer Vision.

ÖZET

ÇEKİRDEK KORELASYON SÜZGEÇİ TABANLI ÇOKLU YAYA TAKİBİ

Ibrahim ABU ARAFEH

Bilgisayar Mühendisliği
Tez Danışmanı: Prof. Dr. Nafiz ARICA

Ocak 2019, 57 Sayfa

Bu tezde Çekirdek Korelasyon Süzgeci (ÇKS) tabanlı nesne takip yöntemi kullanılarak video görüntülerinde çoklu yaya takibi çalışılmıştır. Nesne takibinde veri ilişkilendirme sonuçlarının iyileştirilmesi için Macar algoritması ile birlikte Kesişimin Birleşime Oranı (KBO) yöntemi kullanılmıştır. Önerilen yöntemin performansı, Çoklu Nesne Takibi (ÇNT) Veri kümesindeki ölçütlere göre değerlendirildi. Her iki yöntemin birleştirilmesinin, artan hesaplama karmaşıklığına rağmen nesne izlerinin daha doğru bulunmasına katkı sağladığı gözlenmiştir. Diğer yöntemler arasında, bu araştırma aynı zamanda bazı senaryolarda veri birliği için IOU ile Yapısal Benzerlik Endeksi'ni (YBE) birleştirilerek de uygulamıştır. Özellikle gece senaryolarında, görüntülerdeki parlaklık değişikliklerini kullanan bu model, diğer video dizileri arasında daha iyi genel sonuçlar veren KBO tabanlı yöntemden daha iyi sonuçlar vermiştir. Bu çalışma, çoklu yaya içeren video verisi üzerinde çoklu nesne takibi için sağlam bir görüntü modelinin benimsenmesinin, aynı veri ilişkilendirme yöntemini kullanan hareket modeline göre daha iyi performans gösterebileceğini göstermektedir.

Anahtar Kelimeler: Çoklu Nesne Takibi, Çekirdek Korelasyon Süzgeci.

CONTENTS

TABLES	viii
FIGURES	ix
ABBREVIATIONS.....	x
SYMBOLS	xi
1. INTRODUCTION	1
2. BACKGROUND	5
2.1 TERMINOLOGIES	5
2.2 MOT PROBLEM	6
2.2.1 Problem Formulation	7
2.2.2 MOT Categorization	7
2.3 MOT COMPONENTS	11
2.3.1 Appearance Model	11
2.3.2 Motion Model	17
2.3.3 Interaction Model	18
2.3.4 Exclusion Model	20
2.3.5 Occlusion Handling	21
2.3.6 MOT Evaluation	23
3. LITERATURE SURVEY	25
3.1 VISUAL TRACKING APPROACHES	25
3.1.1 Tracking Using Matching	25
3.1.2 Tracking Using Matching with Extended Appearance Model ...	27
3.1.3 Tracking Using Matching with Constraints	28
3.2 MOT APPROACHES	29
3.2.1 Simple Online and Realtime Tracking (SORT)	29
3.2.2 Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic	30
4. PROPOSED WORK.....	31
4.1 METHODOLOGY	31
4.1.1 Detection	32
4.1.2 Tracking	32

4.1.3	Data Association.....	35
4.1.4	Creation and Deletion of Track Identities	40
5.	EXPERIMENTAL RESULTS	41
5.1	DATA	41
5.1.1	Datasets.....	41
5.1.2	Detections	42
5.1.3	Data Format	43
5.2	EXPERIMENTS	45
5.2.1	Metrics	48
5.2.2	Performance Evaluation.....	49
6.	CONCLUSIONS	56
	REFERENCES	58

TABLES

Table 5.1 :	Descriptions of video sequences used (MOT2015 training)	42
Table 5.2 :	Properties of video sequences used	43
Table 5.3 :	Performance results of the proposed method comparing three methods for data association	49
Table 5.4 :	Detailed performance of all sequences in the MOT2015 training data set of the proposed approach using IOU	50
Table 5.5 :	Performance results of the proposed method comparing three methods for data association	50
Table 5.6 :	Detailed performance of all sequences in the MOT2015 training data set of the proposed approach using IOU	51
Table 5.7 :	Detailed performance of all sequences in the MOT2015 training set of the proposed approach using SSIM only.....	51
Table 5.8 :	Detailed performance of all sequences in the MOT2015 training set of the proposed approach using the mixed IOU&SSIM	52
Table 5.9 :	Performance results of the proposed method on MOT17 training data sets and public detections	53
Table 5.10 :	Performance results of the proposed method on MOT15 training data set and FrCNN private detections.....	53

FIGURES

Figure 2.1 : Detection responses.....	6
Figure 2.2 : DBT and DFT.....	8
Figure 3.1 : Reference model of trackers with the five most important components	26
Figure 4.1 : Flowchart of our algorithm	31
Figure 4.2 : Shifted samples of a base sample that appears in the middle. The translation shift in this figure is done vertically.	34
Figure 4.3 : The circulant matrix illustrated. Each row represents a translated shift of the base sample in 1D.	34
Figure 4.4 : SSIM Compared to Mean Squared Error in Image Similarity.	37
Figure 5.1 : Data Format MOT16	44
Figure 5.2 : Data Format MOT15	44
Figure 5.3 : Labeled objects as seen from the output frames of our method on the DPM dataset.	46
Figure 5.4 : Labeled objects as seen from the output frames of our method on the DPM dataset.	46
Figure 5.5 : Labeled objects as seen from the output frames of our method on the DPM dataset	47
Figure 5.6 : Labeled objects as seen from the output frames of our method on the DPM dataset	47
Figure 5.7 : Published results of trackers submitted online on MOT15 benchmark.	54
Figure 5.8 : Published results of some other trackers on MOT15 benchmark. ..	54

ABBREVIATIONS

DBT	:	Detection Based Tracking
DFT	:	Detection Free Tracking
IOU	:	Intersection Over Union
KCF	:	Kernelized Correlation Filter
MOT	:	Multiple Object Tracking
SSIM	:	Structural Similarity



SYMBOLS

Cost function	:	C
Score function	:	S
Score function tuning factor	:	λ



1. INTRODUCTION

Multiple Object Tracking (MOT) is an important problem in computer vision area. MOT as a whole, includes locating objects in a video sequence, labeling their identities and producing individual trajectories. In previous and current researches, tracked objects included: street pedestrians, sport players in court, flocks of animals, body cells, and others. MOT forms a base ground for higher-level tasks and application in computer vision. Some of these are:

- (a) Visual Surveillance: video surveillance requires automatic analysis for detection of different behaviors. Targets locating and tracking is needed here to obtain the required information, which can be directly achieved in MOT.
- (b) Human Computer Interface (HCI): Visual tracking is required as a basis for extraction of visual information such as expressions and gestures in order to achieve advanced HCI. Therefore, MOT is an important step to make improve HCI and give better results.
- (c) Virtual Augmented Reality (VAR): MOT can be applied in VAR applications and can be used to supply users with the required services.
- (d) Medical Image Processing: MOT can reduce the cost of labeling used for multiple cells in images.

MOT faces many challenges increasingly. Some of these are: frequent occlusions, initializing and termination of tracks, small size of objects, visual similarities and interaction between multiple objects. And although many approaches and researches have been proposed to address the MOT problem, there are still some difficulties that are challenged by many of these researches, or issues that are still unresolved such as sudden appearance changes in the scene and severe object occlusions.

The MOT problem can be defined as a data association problem between detection responses at each frame and the right object identity that they belong to. In order to achieve this, trackers may follow different approaches either considering the motion model Dicle et al. (2013) Yoon et al. (2015) of the object, or the appearance model Bewley et al. (2016c); Kim et al. (2015).

The focus in this research is about pedestrians due to some reasons such as being more non-rigid object in comparison to other objects, and thus being a perfect example for MOT problems. In addition, processing of pedestrians in videos is commercially highly demanded. And thirdly, 70% of current MOT researches are focusing on pedestrians, according to incomplete statistics by Luo et al. (2014). This thesis introduces a solution for tracking pedestrians as part of the MOT problem with an implementation approach of tracking-by-detection, for which the detection process is independent of the tracking process, and the detection responses are taken as input to the tracker.

The motivation of this research is the need to build a simple and accurate tracker which combines the robustness of an appearance model and the simplicity in data associating of several trackers to handle the MOT problem, which has considered the Hungarian algorithm and the intersection over union approach. In this research, we have utilized the image information to enhance performance. Since the existence of an object is known, this can be used to re-locate it by visual correlation. In many previous researches, Bolme et al. (2010) Danelljan et al. (2014) Henriques et al. (2015) visual single-object tracking has achieved important advances by tracking a single object over a long time span starting with the initial object coordinates.

Among many approaches for object tracking, KCF has become a popular and robust tracker due to some reasons, such as being fast and having a simple implementation. Moreover, it has ranked third in terms of accuracy among many other trackers Kristan et al. (2014). Our results have been tested and evaluated using MOT Challenge Benchmark Milan et al. (2016), which is a benchmark that attempts to establish a unified framework towards important quantification of multi-target tracking. Our method was tested

using both the training and testing data sets, and the results were published on the MOT Challenge website in the trackers list running on the 2D MOT 2015 testing data sequences.

In the experiments part, we have compared three ways to handle the data association problem. These were: Intersection Over Union (IOU), Structural Similarity Index (SSIM), and a mixed method of both.

Furthermore, we have compared the results of our proposed method to other methods that adopted the motion model along with similar data association methods to ours (IOU). From these experiments, we have intended to see the effect of utilizing the appearance model instead of the motion model to handle the tracking problem.

In the last part of our experiments, we compared the results of our method to other methods that were tested against the MOT Challenge benchmark. Our results were tested and accepted and can be found in details on the online website of MOT Challenge.

Overall, IOU as a data association method was able to perform better in most of the tested sequences. We refer this to that the use of IOU was able to give more confidence to the tracking results of KCF without further changes in terms of image information, and thus adopting a robust method in handling image similarities which can perform better than other simpler image similarity handling functions.

From night scene sequences, we noticed that the mixed model was outperforming the IOU model in terms of accuracy and the number of tracked objects. We can observe here that the utilization of SSIM to discriminate between close objects was able to solve a problem in night scenes, due to the calculation of the luminance and contrast similarities in the process of data associating tracks to corresponding detections.

Generally speaking, we concluded that our method was able to give the better results in static camera scenes in comparison to scenes with a moving camera, which shows that the algorithm may lose some of its accuracy when applied to scenarios of the later type. We can also notice that when the boxes in the sequence was highly increasing, the overall

accuracy of was decreasing. This can be due to the increasing in similarities between the several boxes which makes the matching and assignment problem harder.

Regarding the problem of short-term occlusions, the IOU distance comparison that we applied in data association was able to implicitly handle these cases that are caused by passing objects. In this case, when a tracked object is covered by an occluder, only the occluder will be detected and matched with the corresponding detection easily, while the occluded will be detected and matched as long as the visible part of it is tracked by the KCF tracker and matching a corresponding detection. When the occluded gets out of the scene it remains unassigned and can get assigned to the same track if it showed up a in a period that is shorter than a certain threshold.

The first chapter of this thesis is the thesis introduction which discusses the motivation of the research and defines the MOT problem, and the focus of this research. The next chapter is the "Background", which explains the MOT problem, its components, models, metrics, evaluation and then discusses related studies to the work and their relationship. In the "Literature Survey" chapter, we give an overview of the basic approaches that addressed the visual tracking in general and then the MOT problem in specific. The third chapter is the Proposed Work's chapter, which states the components of the proposed method for solving the MOT problem, and illustrates its results as measured by the MOT benchmark Leal-Taixé et al. (2015). The last chapter is the "Experimental Results", which illustrates the results of running the proposed method on several data sets and explains them. Finally, summarizing the conclusions and future work of this research in "Conclusions" chapter.

2. BACKGROUND

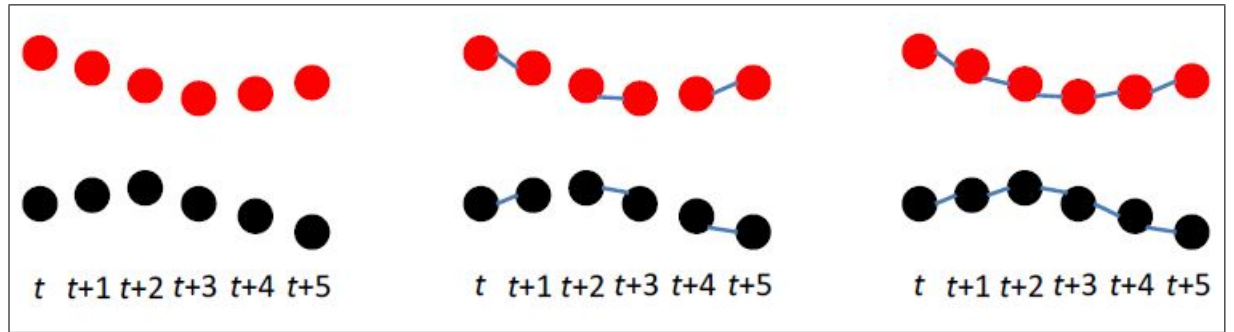
This chapter explains the background of multiple object tracking. The content of this chapter is divided into a list of sections, which are: Terminologies, MOT Problem and MOT Components.

2.1 TERMINOLOGIES

Following is a list of important terminologies in object tracking in computer vision that plays significant role in MOT research.

- (a) **Object:** In computer vision, an object is considered as a continuous closed area in an image which is distinct from its surroundings. MOT studies are usually interested in a certain type of object, thus, an object has to be further identified with another identity such as pedestrian, animal, cell, etc.
- (b) **Detection:** Detection is a computer vision task which localizes objects in images. An object detector is usually trained using a training set in order to conduct this. In most cases, temporal information is not included in detection.
- (c) **Tracking:** Tracking is to locate an identical object in a set of consequent frames. In MOT, it means simultaneously localizing multiple objects and maintaining their identities. Temporal information is always involved in tracking.
- (d) **Detection Response:** is the output of the detection process. This is also the output of a trained detector model for a certain type of objects, which could be pedestrians, cars, cells, animals or others. It is also known as detection observation or detection hypothesis. It also configures the object in an image sequence in terms of position, size, etc.

Figure 2.1: Detection responses and how they combine to form trajectories.



Source: Luo et al. (2014)

- (e) **Trajectory:** represents the output of an multiple object tracking process. A trajectory is unique, because each trajectory corresponds to one target. One trajectory is composed of a set of object responses for an identical target in a video sequence. Each object response contains the relevant information to uniquely identify the object and its spatial and temporal information. Figure 2.1 illustrates trajectories and detection responses.
- (f) **Tracklet:** a tracklet comes in between a detection response and a trajectory. It is a set of detection responses that belong to an identical target. A tracklet is shorter than a trajectory in time span and is formed of confident detection responses linked together. Detection responses can be linked into much longer tracklets in order to shape and produce trajectories as a final step.
- (g) **Data Association:** is considered as a paradigm of matching detection responses across frames based on object detection.

2.2 MOT PROBLEM

The main objective of MOT is to produce trajectories of objects as they move around the image plane. In this section we are going to view the formulation of the MOT problem and types of MOT categorization based on different criteria.

2.2.1 Problem Formulation

For an input of image sequence $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_t, \dots\}$ then s_i^t is employed to indicate the state of the i – th object in the t – th frame. To estimate the states of objects, we need to collect some observations from the image sequence.

The objective of multiple object tracking is to find the “optimal” sequential states of all the objects, which can be generally modeled by performing (maximal a posterior) estimation from the conditional distribution of the sequential states of all the objects given all the observations.

2.2.2 MOT Categorization

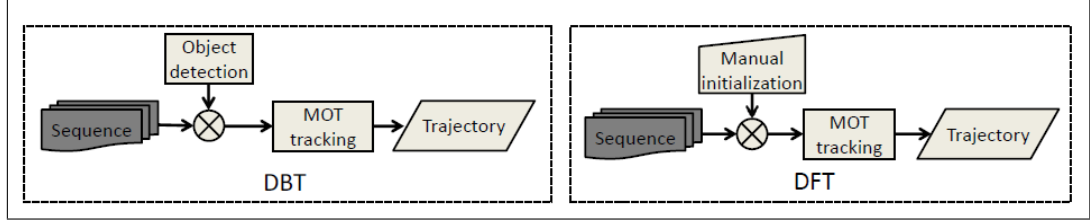
It is not easy to get a unified classification for all MOT problems. However, an MOT problem can be categorized according to different criteria to reach a better understanding of the problem. Following are some aspects to provide discussion of famous MOT problems.

2.2.2.1 Initialization Method

In this criterion, solutions are categorized on the basis of how objects are initialized in them. Most MOT work is grouped into two sets: Detection Based Tracking (DBT) and Detection Free Tracking (DFT). DBT depends on object detection while DFT does not.

- (a) DBT Objects are first detected in frame, and after that object hypotheses are linked into trajectories through sequential or batch tracking. In DBT, there are some issues need to be considered. First, detection procedure is not a focus in DBT, as most cases a pre-trained model is used to generate detection responses as input. Moreover, DBT researches target a certain type of objects, such as humans, cars, animals or others. In addition, the detector model’s performance can have a major impact and

Figure 2.2: Procedure flow of DBT (left) and DFT (right)



Source: Luo et al. (2014)

role in the performance of the DBT.

- (b) DFT starts with manually initializing a set of objects in the first frame which will be localized in the later frames. Initialization usually takes place in the form of a bounding box or other shapes. DFT does not need an object detector model to achieve its task. Comparing both approaches, DBT is being more popular due to the fact that it is more practical in detecting new objects dynamically and that disappearing objects are terminated automatically. DFT on the other side requires manual initialization of each object, and so cannot help when some objects show up lately. As DFT is free of detection models, it can handle videos of objects of any kind. But the setting of fixed number of objects limits its practical use in applications. Figure 2.2 illustrates the difference between DFT and DBT models.

2.2.2.2 Processing Mode

In perspective of processing data, the tracking categories are online and offline. This differs whether future frame observation is going to be processed when handling the current frame. Online tracking uses frame observations up to current frame only, whereas offline tracking uses past and future observations.

- (a) Online Tracking. Image sequence is handled in step-wise way, so it is also called sequential tracking.

- (b) **Offline Tracking.** It utilizes a batch way to process data, so it is also called batch tracking. Observations from all frames are required in advance during the processing of the current frame. In some cases, due to performance limits handling all frames together is not possible, and therefore, video could be divided into a set of segments and each segment would be being processed separately and the final result is fused at the end hierarchically. Online tracking would be more suitable in a case where video is obtained sequentially. Offline tracking deals with data globally when all data frames are obtained.

DFT may seem look identical to online tracking as it always processes observation sequentially. On the other hand DBT may look similar to offline tracking, since tracklets and detection responses in DBT are connected in a batch way.

2.2.2.3 Mathematical Methodology

MOT could also be categorized into probabilistic and deterministic tracking according to the adopted mathematical methodology. In probabilistic tracking, estimation is based on probabilistic inference. Whereas in deterministic, estimation is based on deterministic optimization. Furthermore, output of probabilistic tracking may be different in different running trials while constant in deterministic tracking.

2.2.2.4 MOT Special Cases

The following cases cannot be easily categorized as traditional multi-object tracking:

MOT in sport scenarios: MOT could be used in several applications in sport scenarios, such as for match statistics and analysis of strategies. Traditional MOT is different from this problem for several reasons. First, sport court is special in itself. Recording process includes several zoom ins and outs and frequent shots and switches. Moreover, the specialty of the court could also be helpful as it forms certain boundaries which differentiate

the area of interest from other areas. In some research Xing et al. (2011), it has been proposed to separate the playfield from the non-playfield as an initialization. Secondly, players from the same team would be wearing the same uniform. This could make harder than traditional MOT cases to differentiate players of the same team. But, on the other hand, this could be helpful to train a classifier to classify team players using features such as RGB color histograms for examples as in Lu et al. (2013).

MOT in aerial scenes: MOT in aerial scenes has some features that make it different from normal MOT. For example, in such scenes objects tend to look much smaller, which makes the appearance model not the best choice in terms of reliability. Moreover, sequences of such scenes would low frame rates, and high object density.

Generic MOT: as mentioned previously, the majority of MOT work has focused on certain type of objects. For example, humans or cars. And since these methods are mainly dependent on object detector models that was a reason for DBT solutions to be more popular and practical in real-life applications. However, there are also some problems. First, a trained detection model can handle and run on video sequences of the same kind of objects. Second, these detection models are data-specific, and so the performance may reduce when they are applied to other images sets, whereas DFT solutions can be applied to type-free sequences although it requires manual initialization.

To address the previous problems, many researches have been made to make the MOT problem generic for any type of objects with minimum manual labeling and free of offline trained models. In Zhao et al. (2012) one instance is to be labeled in the first frame. Then, this object is tracked and training samples are collected as training data for a detection model for this type of objects in the early frames. Later, a new detection starts again from the beginning of frames sequence in order to detect the top M objects with similarity and then those will be tracked in the upcoming frames. This proposal can make a good save in labeling cost in comparison to DFT. But, since the number of similar object is fixed, it is still not comparable to DBT in generality to object types. Object detection is gradually developed by accumulating training samples. Once an object is discovered, a specialized

tracker is initialized and multiple trackers are learned together based on multiple task learning.

2.3 MOT COMPONENTS

MOT involves two main components: observation model and dynamic model. Observation model measures similarity between object states and observations. It includes modeling of appearance, motion, interaction, exclusion and occlusion. Dynamic model studies state transitions across frames. It is categorized into probabilistic inference and deterministic optimization.

2.3.1 Appearance Model

Appearance plays a significant role in multiple object tracking, but it is not the main focus of tracking, as much as it is in the case of single target tracking. This is due to the fact that, in MOT, multiple objects can be hardly discriminated based on merely the appearance information. Appearance model contains two main components, visual representation and statistical measuring. In visual representation, object is being described precisely in terms of image features and visual characteristics. This is divided into: visual representation based on single cue, and visual representation based on multiple cues. On the other hand, statistical measuring is about computing how much similar or dissimilar observations are when representation is ready. Appearance modeling can be illustrated in the following equation:

$$S_{i,j} = F(o_i, o_j), \quad (2.1)$$

where $F()$ is a function to measure the similarity S between o_i and o_j , where o is a visual representation of some observation.

2.3.1.1 Feature

Features are totally necessary for MOT. They are categorized into the following subsets.

- (a) Point features. In MOT, point features can be helpful sometimes. For example, in KLT tracker feature points are being tracked in order to produce a sequence of short tracklets or a sequence of trajectories.
- (b) Color (intensity). It usually comes along with another measurement to calculate the affinity between two counter-parts. It is also the most widely used in MOT.
- (c) Optical flow. The use of this feature could apply in short-term object tracking, where several MOT approaches utilize optical flow to link detection responses from consequent frames into trajectories or tracklets in the data association process. Moreover, this feature is widely used in hugely crowded scenes for detecting motion patterns of the crowds.
- (d) Gradient/ pixel-comparison features. A variation of the level-set formula has been utilized by Mitzel et al. (2010). Three terms are integrated to penalize the deviation between foreground and background.
- (e) Region Covariance Matrix. Region Covariance matrix has very good results dealing with illumination changes and scale variations, such has been used in Porikli et al. (2006).
- (f) Depth. Depth is used in multiple computer vision tasks. In Ess et al. (2009), depth information was employed to flag detection hypothesis with 0 or 1 for depth flag, which refines the detection responses.
- (g) Others. Some other features which are less popular were also used to conduct MOT. For example, gait features in frequency domain could be unique for every person, and can be helpful to increase the discrimination between tracked individuals, as employed by Sugimura et al. (2009). The periodic component of a trajectory could

be extracted by fitting a linear regression line in a trajectory. By applying Fast Fourier Transform to the residual periodic signal, the amplitude spectra and phase of trajectory are obtained. Then the dissimilarity between trajectories can be computed.

To sum up, every feature has its own efficiency as well as its own shortcomings. For example, color histogram is good in similarity measures, but lacks spatial information of the object region. Point features are efficient in tracking, but cannot deal well with occlusion or other scenarios such as out-of-plane rotation. On the other hand, HOG and other gradient based features would give a good description of the object's shape and are strong against illumination changes. However, they are strong against occlusion and deformation. Region covariance matrix is robust since it obtains enough information but has a high cost in performance instead. Depth features are accurate in affinity computation but need different views of the same scenery.

2.3.1.2 Single cue based appearance model

Single cue is a popular option in MOT. Following are appearance models that employ this in five different ways.

- (a) Raw pixel representation. The pixel can be the most popular element in computer vision problems. Its representation can be as a region's color, or else can be as the intensity of the raw pixel. The pixel representation holds information of spatial category, where it compares the matching between two targets in element wise. In some approaches, the object's potential location was calculated using Normalized Cross Correlation as in Yamaguchi et al. (2011). In other approaches, the Normalized Cross Correlation is calculated between the current target and a candidate's bounding rectangle to find the appearance affinity. Wu et al. (2012) network-flow method is proposed to deal with MOT. The transitional cost on the arc of the network is computed as flows, then the NCC is used to compare between the upper quarter

bounding rectangles of the detection responses. This representation is efficient in general but has some drawbacks in case of occlusions or change in illumination.

- (b) Color histogram representation. It is a very popular representation of appearance models in MOT. This refers to its capability to effectively obtain the statistical information of target region. In some approaches like Kratz & Nishino (2010a) the color histogram representation has been utilized to compute the likelihood in terms of appearance, and then the histogram distance is transformed into probability using an exponential function. In other approaches, the dissimilarity is being captured using the Bhattacharyya distance and the histograms of hue-saturation, as has been applied in Sugimura et al. (2009). In Leibe et al. (2008), the histogram of RGB color in each trajectory has been used as a representation of the appearance model. The color histogram of the first detection response is initialized and develops as an average of the detection observations of this trajectory. The probability of appearance is proportional to the Bhattacharyya coefficient of two histograms. Affinity of appearance is calculated as Bhattacharyya distance between the mean HSV color histograms of the concerned tracklets. In Xing et al. (2009), a demonstration of pedestrian body as has been made presenting it in three overlapped parts. Full Body , Head-Shoulders , and Head-Torso, which were denoted as (FB, HS, HT) respectively. The appearance affinity between two tracklets T_i and T_j is:

$$S(T_i, T_j) = \exp(-D(c_i, c_j)), \quad (2.2)$$

where $D(c_i, c_j) = \text{mean}\{D(c_i^k, c_j^k) | k = HT, FB, HS\}$, c denotes color histogram where $D()$ denotes the measure of Bhattacharyya distance.

On the other hand, the representation of color histogram has the shortcoming of missing spatial information.

- (c) Covariance matrix representation. It has proved its effectiveness against changes in rotation or illumination. The object appearance is represented by a covariance matrix descriptor. The probability concerning appearance linking two detection

responses is given as:

$$P_{link}(T_i, T_j) \propto N(D_{ij}; \bar{D}, \Sigma), \quad (2.3)$$

N denotes Gaussian distribution, D_{ij} denotes the dissimilarity in appearance between T_i, T_j , $\bar{D}$ and Σ are parameters estimated from training data.

- (d) Pixel comparison representation. Could be the simplest representation, in which two pixels are being compared to produce a result of binary value. It can use training data in the comparison process to get the comparison results encoded between pixel pairs.
- (e) Bag of words representation. It computes SIFT features and uses the bag of word to encode them, as has been used in Lowe (2004). The spatial information could be included by using the spatial pyramid matching SPM method.

2.3.1.3 Multi-cue based appearance model

Multiple cues could be used together to make the model robust. The challenge here would be how to combine the cues and fuse them in the right way. To do so, five kinds of fusion strategies are proposed.

- (a) Boosting. Boosting is a way to select multiple features to be used in the appearance model. The features are being selected sequentially from a pool of features using some Boosting algorithms, such as Adaboost or RealBoost, Luo et al. (2014).

Tracklets with high similarities are being assigned to same objects, where those with lower similarities are assigned to different ones. The process takes place using a discriminative appearance model that gives such suggestions. A discriminative appearance model is suggested to assign high similarity to tracklets of the same object, and low affinity to tracklets of different objects. 15 regions are selected and 45 cues are produced in total in the feature pool. Later, Adaboost is used to select

the most representative features to distinguish pairs of tracklets of the same object from those of the different objects. Kuo et al. (2010)

- (b) Concatenating. Target is discriminated from other targets using a trained SVM model classifier. Features that were used in concatenation included histogram of oriented gradients, color and optical flow. Later, dimension reduction was applied using a PCA project in order to describe the detection response.
- (c) Summation. Multiple objects are simultaneously segmented and tracked. In pedestrians' case, color information alone may result in unreliable segmentation. A proposed solution is to integrate color information with depth information. At first, an expected depth is calculated from the foreground of the next frames based on depth and velocity. Later, a Gaussian distribution is used to assign a probability for every depth in the next frame. On the other side, the probability of the color feature is corresponding to the color histogram. The two probabilities are then multiplied by a parameter (α) which acts as a weighting value: (2.4).

$$P_i = (1 - \alpha)P_{i;color} + \alpha P_{i;depth}; \quad I \in f_g; \quad b_g \quad (2.4)$$

- (d) Product. Multiple cues such as color, shapes and bags of local features are integrated to find the probability of associating a detection observation into a current trajectory. The probability of linking a detection response with a trajectory (stat s) is : $P(f^1, f^2, f^3|s)$. This probability is calculated as the product of their probabilities of (f^1, f^2, f^3)

$$P(f^1, f^2, f^3|s) = \prod_{i=1}^3 P(f^i|s). \quad (2.5)$$

- (e) Cascading. In Gavrilu & Munder (2007) depth, shape and texture are used in cascade way to limit the searching domain of multiple target tracking. Hypotheses are generated first using the depth feature. Later, texture and shape were applied sequentially. Depth is also used again to verify detections.

2.3.2 Motion Model

Object motion model describes the movement of an object. It estimates the potential location of the object in the next frames which reduces the search space. Motion models in MOT are categorized into two classes.

(a) Constant velocity/linear motion models

Objects in this model are assumed to move in a constant velocity. The velocity of object in the current frame is equal to the previous velocity (in addition to noise in process).

A linear velocity motion model has been proposed by Breitenstein et al. (2009):

$$(x, y)_t = (x, y)_{t-1} + (u, v)_{t-1} \bullet \Delta_t + \epsilon_{x,y}, \quad (2.6)$$

$$(u, v)_t = (u, v)_{t-1} + \epsilon_{u,v}, \quad (2.7)$$

(x, y) represents the image position in 2D, and (u, v) represents speed. $\epsilon_{x,y}$ and $\epsilon_{u,v}$ are noise variables in Gaussian distributions with the mean of zero. Other constant velocity models are Andriyenko & Schindler (2011) and Milan et al. (2014). A term which computes differences in velocities of one object across different time moments is:

$$C_{dyn} = \sum_{t=1}^{N-2} \sum_{i=1}^M \|\mathbf{v}_i^t - \mathbf{v}_i^{t+1}\|^2, \quad (2.8)$$

where $\mathbf{v}_i^t$ is the velocity of target i at time t . C_{dyn} is the displacement between object positions in two sequenced frames. Then first term of summation takes all the N frames into account and the second counts all the M trajectories/ objects. In general, the term forces trajectories to be smooth by penalizing the difference between velocities. In Xing et al. (2009) it has proposed a linear motion model that considers the forward and backward velocities to calculate the motion affinity between two different tracklets. For instance, assuming we have two tracklets T_i and T_j , and there is a temporal gap between the head of T_i and the tail of T_j . A

Gaussian distribution model is used to describe the forward direction motion. The center of Gaussian distribution is centered at p_j^{head} , the position of the head response in T_j , and the variance is Σ_B^j . The likelihood of the position of tail response in T_i is estimated as forward displacement as $p_i^{tail} + v_F^i \Delta t$.

In Yang & Nevatia (2012), Conditional Random Field (CRF) model has been employed, where each node is represented as pair of tracklets. The node label indicates if the two tracklets of this node are able to be associated or not. The unary term of CRF model is used for that, taking in consideration both the appearance and motion information. The likelihood considering motion is computed according to the displacement between the predicted positions using a constant velocity model and the positions that have been observed.

(b) Non-linear motion model

In some cases where linear motion model cannot deal with movement of the object, non-linear motion models are suggested to produce more accurate affinity between tracklets. In Yang & Nevatia (2012), for two tracklets T_1 and T_2 that belong to the same target, the linear motion model would produce low likelihood to connect them, which is far from the reality. Using the non-linear motion model, the gap between tail of tracklet T_1 and head of tracklet T_2 could be explained by tracklet $T_0 \in S$. Tracklet T_0 is a support tracklet to explain T_1 and T_2 because there exist elements (position, size and velocity) in T_0 which are matched with the head of T_2 and the tail of T_1 . The real path to bridge T_1 and T_2 is estimated based on T_0 .

2.3.3 Interaction Model

Also known as mutual motion model, detects the influence of an object to other objects. Interaction model can be divided into two models: models of social force, and models of crowd motion pattern. For social force models for example, speed and direction are consideration taken by a pedestrian to avoid collision with other objects. Crowd motion pattern models, study the motion of crowded scene, including the moving objects and the

environment factors such as barriers, paths and crowd motion. The whole crowd would form a motion pattern. The two models are illustrated as follows.

(a) Social force models

Every object depends on other objects and environmental factors. In this type of models, observations of other objects are used along with the surrounding environment in order to determine the speed, direction and destination of objects. In addition, we can categorize these models into two aspects:

- (i) Individual force. Individual force implies that a moving object is not expected to suddenly change its velocity (speed and direction), as well as not being expected to suddenly change its destination.
- (ii) Group force. In group force, a group moving together should stay close to each other. In addition, they should stay away from each other such that they are moving comfortably. Finally, a group of moving individuals should move with a similar velocity.

For each pedestrian, in order to avoid possible collisions Pellegrini. (2009) maximizes the minimum distance which makes targets comfortable. For this distance, there is a corresponding energy term C_{ij} between each two targets (i, j) . Every energy term is balanced by a parameter w representing weight, and so the energy between the object i and other objects considering interaction would be equal to:

$$C_i^{inter}(v_i) = \sum_{j \neq i} w_{ij} C_{ij}(v_i), \quad (2.9)$$

Scovanner et Tappen. (2009) have proposed a social force model with four components to describe movements of pedestrians in real scenarios. The model which consists of four components that correspond to energy terms can be viewed as follows: C_{LM} , constraints the motion of an object to keep from jumping in the space grid. C_{cv} , maintains the object's constant speed. C_{Dest} , guides target to destination.

C_{AV} takes other targets into consideration, and produces repulsion to avoid possible collisions.

(b) Crowd motion pattern models

This type of motion models are helpful in highly crowded scenes. An object can be tracked more easily by taking the crowd's factors and environment into consideration. This is also much helpful in scenes with high targets density where the appearance model becomes less contributing due to the high appearance similarity among targets.

In crowd motion pattern models, three types of forces fell into category. First, Static Floor Field (SFF) which deals the scene structure factors such as a favorite path of the crowd and exit points. Secondly, Boundary Floor Field (BFF) which deals with physical and virtual scene's barriers and how identify them to help the tracking process. Lastly, there is the Dynamic Floor Field (DFF) which deals with the potential motion and positions of the crowd that is surrounding the tracked object.

As in Zhao et al. (2012), crowds demonstrate obvious motion forms for MOT cases. The motion patterns can be discovered by an ND tensor voting of tracklet points from KLT tracker. A 4-D points model is considered in order to represent the motion patterns of spatial positions and speeds along 2 dimensions. These patterns are then utilized in order to compute the likelihood of a detection candidate, and to estimate the speed of a target in a particular tracking position.

2.3.4 Exclusion Model

Exclusion could be a constrain in MOT due to physical collisions. In general, there are two exclusion constraints: 1) detection-level exclusion, which is two different detection responses cannot be associated to same trajectory hypothesis in the same frame, 2) trajectory-level exclusion, that is two trajectories cannot occupy an identical detection response.

(a) Detection-level exclusion modeling

This is illustrated by defining a cost term to add a penalty when two simultaneous detection responses at time t are assigned to the same trajectory. A particular graph of exclusion is built to detect the constraint that simultaneous detection responses should have different labels. In this graph, nodes represent all detection responses, and are not fully connected. Every node is only connected to simultaneous nodes, and the connecting weight of each edge is uniform as $w_{ij} = 1/n$, where n is the number of detection responses that occur at the same time as these two nodes. Later the Laplacian matrix of the graph is calculated as L and label error of exclusion is maximized as,

$$\arg \max_{Y \in S} Tr(PLY), \quad (2.10)$$

(b) Trajectory-level exclusion modeling

This is modeled with an energy function as:

$$C_{exc} \propto \sum_{t=1}^N \sum_{i \neq j} \frac{1}{||p_i^t - p_j^t||^2} \quad (2.11)$$

The first summation takes all N frames in the video, where the second summation takes all the detection responses in a frame. The energy cost would be apparently high if the detection responses are too close.

2.3.5 Occlusion Handling

Occlusion is one of the significant problems in multi target tracking. It may result in ID switch or fragmentation of trajectories. To deal with this, following are the popular methods in this aspect.

Part-to-whole, in this most popular strategy part of the object is still visible during occlusion. This is assumed in most cases since even a complete occlusion begins with a partial occlusion. The visible part is used to conclude state of the whole object.

Hu. (2012) proposes a block-division model to treat occlusion. Object is firstly divided into many non-overlapped partitions. An appearance model which uses subspace learning is built for each division. The advantage of this model is that an occlusion map is obtained from reconstruction errors of all partitions, and this map is used to conclude the occlusion relationship among object, which could be later used to update the appearance model.

Tracking according to appearance model may fail if occlusion happens. Beside part based model, feature point clustering based tracking can also be used to address occlusion. In this proposal, feature points with high motion similarity are supposed to match with the same target. The resulted trajectories will work as long as some parts of the target are visible.

Hypothesize-and-test. In this strategy some proposals are hypothesized and tested later according to observations at hand. Zhang. (2008) propose an Explicit Occlusion Model (EOM) integrated into the cost low framework to handle long-term occlusion in data association for MOT. For instance, we can link two tracklets if the time distance between them is relatively small. If the temporal gap threshold is increased, this would make more tracklets linkable and would make the situation easier but may also produce stray associations. To address this, occlusion hypotheses would be generated according to the occlusion constraints. If the distance and sale difference of two observations are small enough then they are occludable. Meanwhile, occlusion could also be used to help detection by producing typical appearance patterns. For example, a double-person detector is modeled to detect different levels of occlusion between two pedestrians.

Buffer-and-recover, it buffers observations at the moment of occlusion and stores the states of objects before occlusion. After occlusion, object states are restored according to the buffered observations and the stored states before occlusion. Mitzel et al. (2010) integrate a level-set tracker based on image segmentation. A high-level tracker is used to start with new tracks from detection response and the level-set tracker is employed to treat the frame-to-frame data association. When occlusion takes place the level-set tracker would fail. Therefore, the high-level tracker keeps a trajectory alive for up to 15 frames

and predicts the position to extend the inanimate trajectory through occlusion.

2.3.6 MOT Evaluation

2.3.6.1 Metrics

MOT evaluation metrics are crucial by providing standard for fair quantitative comparison. Most approaches take into consideration the measurement of performance for both detection and tracking.

- (i) Metrics for detection These are classified into two subsets: accuracy, and precision.
 - (a) Accuracy. It commonly uses Recall and Precision metrics in addition to the average False Alarms per Frame (FAF) rate. Yang. (2011).
 - (b) Precision. The quality of alignment between true detections and the ground truth is measured with the Multiple Object Detection Precision (MODP). Kas-turi. (2009).
- (ii) Metrics for tracking.

Tracking metrics can be categorized as follows:

- (a) Accuracy, this metric measures how accurately an algorithm could track objects. The metric of ID switches counts how many times a MOT algorithm switches to wrong objects. Yamaguchi et al. (2011)
- (b) Precision, it describes how precisely the objects are tracked from the view of overlap and distance, using Multiple Object Tracking Precision and Tracking Distance Error.
- (c) Completeness. These metrics indicate how completely the ground truth trajectories are tracked. Metrics of Mostly Tracked (MT), Partly Tracked (PT) Li. (2009).

- (d) Robustness, assesses the ability of a MOT algorithm to recover from occlusion. Recover from Short-term occlusion (RS) and Recover from Long-term occlusion (RL). Song. (2010).



3. LITERATURE SURVEY

This chapter talks about the literature survey of the research. The content of this chapter is divided into a list of sections, which are: Visual Tracking Approaches and MOT Approaches. We are going to state some of the solutions and studies that addressed the problem of object tracking, and multiple object tracking in general.

3.1 VISUAL TRACKING APPROACHES

In this section we are going to review some of basic approaches to handle the visual/object tracking problem before we specify in the MOT approaches. The approaches explained in this part were taken from a Visual Tracking Experimental Survey by Smeulders et al. (2014) which has evaluated 19 trackers systematically and experimentally on 315 video frames. Trackers were evaluated by survival curves, Kaplan Meier statistics, and Grubs testing. A total of 315 video fragments were gathered focusing on one situation per video in order to evaluate trackers' robustness.

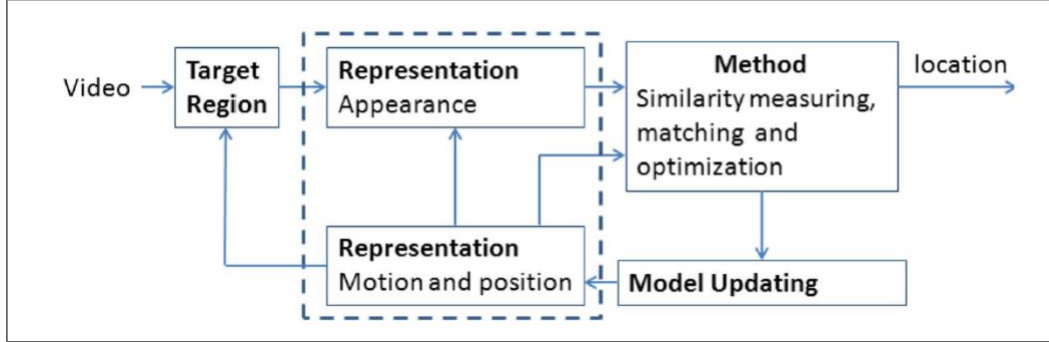
This survey was not generic for all tracking cases but considers what comes under the following definition, that tracking is the analysis of video sequences for the purpose of establishing the location of the target over a sequence of frames (time) starting from the bounding box given in the first frame.

The reviewed visual trackers can be categorized into the following parts:

3.1.1 Tracking Using Matching

The trackers in this group match the representation of the target model built from the previous frame(s). Figure 3.1 illustrates the most important components in visual tracking.

Figure 3.1: Reference model of trackers with the five most important components



Source: Smeulders et al. (2014)

- (a) (NCC) Normalized Cross-Correlation: a basic concept of tracking, which is to match the normalized cross-correlation of the intensity values of the initial target bounding box as template. At each frame, candidate windows are proposed around the target. And each candidate window is compared against the target template using NCC. The candidate with highest score is considered to be the next target location.
- (b) (KLT) Lucas Kanade Tracker: the tracker compares the affined-transformed match between the target bounding box and candidate windows around target. To compute the affine transformation, incremental image was aligned based on spatiotemporal derivatives and warping capable of dealing with scale, translation and rotation. The location of the new target is determined by matching the target position in the previous frame with that in the current frame using affine transformation.
- (c) (KAT) Kalman Appearance Tracker: addresses changes in appearance by appearance-predicted matching for cases when target comes under occlusion. Target is represented as 20x20 template-intensities, each of those has a separate Kalman filter. Using an additive Gaussian noise model, the filter is considered to predict the behavior of each template-intensity over time. A 2-D translation is used to model target motion at a single scale and to search in a region around the previous location of target. Candidate windows in the next frame are reduced to 20x20 templates

and compared against the predicted position. The candidate that has the lowest overall difference would be the next target position. The template here would be updated with the predicted new target.

- (d) (MST) Mean Shift Tracking: a famous tracker that performs matching with histograms without using spatial information about the pixels. This method is suitable for radical shape changes of the target. Target is represented by an RGB-color histogram. In the next frame, candidate windows histograms are compared against that of the target position using Bhattacharyya metric. Mean shift is later used to maximize find the mode of a function, the one that maximizes the Bhattacharyya distance is selected as the next target location. MST does not update template during tracking.
- (e) (LOT) Locally Orderless Tracking: the tracker performs comparison in object appearance by matching with flexible rigidity. In the initial bounding target box, target is divided into super pixels. Each super pixel is to be represented by the center of mass and average (Hue, Saturation, Value) HSV-values. Each target state is sampled using a Particle Filter, and a Gaussian weight is proposed around target's previous position. Each particle represents a candidate window, and the probability of each window is taken from a parameterized Earth Mover's Distance between the target windows and the candidate super pixels. The probability-weighted sum over all windows is selected as the new target state.

3.1.2 Tracking Using Matching with Extended Appearance Model

Stepping forward in tracking using matching, a new idea was proposed which is maintaining an extended model of the target's appearance over the previous frames. This includes searching for the best match in the image and in the extended model of appearance variations.

- (a) (IVT) Incremental Visual Tracking: This tracker captures the full range of appear-

ance of the target in the past to keep an extended model of appearances. Incremental PCA is computed over the target's intensity-value template to compute eigen images of the target, which are stored in a leaking memory to slowly get rid of old observations. Gaussian distribution around the previous position is used as a motion model to sample candidate windows by Particle Filtering. The likelihood of each sample is calculated as the distance of the intensity feature between candidate window and target's Eigen image subspace. The window with the minimum score is selected as the next target position.

- (b) (TAG) Tracking on the Affine Group: using an extended model of appearances, this method extends the motion types (translation, scale, rotation) to a general 2-dimensional affine matrix group. This tracker samples all possible transformations of the target from the affine group using a Gaussian model.
- (c) (TST) Tracking by Sampling Trackers: As the real-world changes considerably over time, the tracker would be required to adapt to the current situation. This method relies on tracking by sampling many trackers, and preserves an extended model of trackers. Four components make each tracker: appearance model, motion model, state representation model, and observation model. Each of them is sub-divided into sub-components. Target state stores the center, scale and spatial information which is subdivided according to vertical projection of edges. Sparse incremental PCA with leaking of (HSI Hue, Saturation, Intensity) features captures the state's appearance past over the last five frames, quite similar to IVT.

In next frame, the basic tracker with the best target state is selected from the space of trackers.

3.1.3 Tracking Using Matching with Constraints

A recent development in tracking which reduce the target representation to a sparse representation and performs sparse optimization. It has got benefit of the successes of sparse

representation in object detection and classifications.

3.1.3.1 Tracking Using Discriminative Classification

In this tracking, the model is built by distinguishing the target foreground against the background. It creates a classifier to discriminate between target pixel and the background one. Then, new coming samples are used to update the classifier.

3.2 MOT APPROACHES

In this part we are reviewing some the similar approaches to our work that have addressed the MOT problem.

3.2.1 Simple Online and Realtime Tracking (SORT)

SORT is one of the similar approaches that addressed the multiple object tracking problem for pedestrians particularly. SORT is also following the tracking-by-detection paradigm, in which all the detections are found prior to the tracking process, and then provided as input to the tracking algorithm in the form of coordinates data. In the tracking process, SORT is adopting the motion model and the Kalman Filter tracker to predict the position of every tracked object in the next frame.

In detection phase, they have used the Faster Region CNN (FrRCNN) detection framework with a trained model for detecting pedestrians.

In tracking phase, the Kalman filter model was applied to estimate the displacements of target between frames with a linear motion model. The next state of each object was modelled as:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T, \quad (3.1)$$

where u and v represent the horizontal and vertical pixel location of the object's center, whereas s represents the scale area, and r represents the aspect ratio of the bounding box. When a new detection is associated to a certain tracked object, the object state would be updated by the detection's bounding box that has been associated to. In data association phase, they have used intersection-over-union (IOU) to compare the tracking outputs with the detections in each frame, along with Hungarian algorithm to solve the cost matrix. SORT has been tested and evaluated using the datasets of MOT15 benchmark Leal-Taixé et al. (2015) and the results were submitted online on the MOT Challenge website. Compared to our work, our main difference with SORT is that by adopting the appearance model and a robust tracker like KCF, we have managed to achieve a better overall accuracy than that achieved by the the motion model and Kalman filter. The results are shown and further explained in chapter 5.

3.2.2 Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic

This research has used Kernelized Correlation Filters (KCF) in multiple object tracking in urban traffic. Since it has originally targeted urban traffic, the research was specific to pedestrians tracking, but was rather tracking all moving objects in the scene like cars and bicycles. This problem was different from the problem that we are addressing, which is only concerned with moving pedestrians. In this method, the algorithm does not use any pre-trained model to detect objects, neither does it use any input detection prior to the tracking process. Instead, the algorithm detects each new or existing object using the blobs produced from running background subtraction method on all frames, which makes the detection and tracking running both at the same time of processing each new frame. In data association phase, the algorithm used the output of KCF tracker to match with the overlapping output of existing trackers, without other methods for data association.

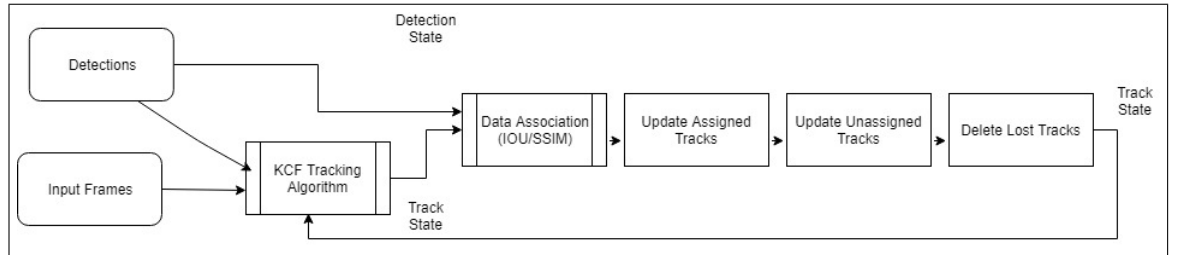
4. PROPOSED WORK

In this research, we have utilized the image information to enhance performance. Since the existence of an object is known, this can be used to re-locate it by visual correlation. In many previous researches, Bolme et al. (2010) Danelljan et al. (2014) Henriques et al. (2015) visual single-object tracking has achieved important advances by tracking a single object over a long time span starting with the initial object coordinates. Among many approaches for object tracking, KCF has become a robust and popular tracker due to its fast performance and simple implementation. Moreover, it has ranked third in terms of accuracy among many other trackers Kristan et al. (2014).

4.1 METHODOLOGY

In this section we explain our proposed method and discuss our implementations in addressing the MOT problem through the multiple phases of detection, tracking and data association. Figure 4.1 illustrates the flowchart of our proposed method.

Figure 4.1: Flowchart of our algorithm



4.1.1 Detection

Several detections were used to run the experiments of the research based on the experiment and the benchmark it was compared against. Those detections included: 2D MOT 2015 public detections, MOT 2017 public detections, and a private CNN based detector using the Faster Region CNN (FrRCNN) detection framework.

- (a) 2D MOT 2015 Public Detections. The detections in this part were the output of the object detector implementation of Dollár et al. (2014b), which is based on aggregated channel features (ACF). The pedestrian trained model was used with default parameters and was trained on the INRIA dataset Leal-Taixé et al. (2015). The model was re-scaled to a factor of 0.6 to enable the detection of smaller pedestrians.
- (b) MOT 2017 Public Detections. The public detections provided for the MOT 2017 data sequences were generated using several detectors, each having its own sequences. These were Scale-Dependent Pooling (SDP) by Yang et al. (2016) which uses deep neural networks. The other detector was FrCNN by Ren et al. (2015).
- (c) Faster Region CNN (FrCNN). FrCNN is a Convolutional Neural Network based detector. The detector is an end-to-end library that is built of two phases. In the first one, features are being extracted, then regions are proposed for the second phase which is supposed to classify the target in the suggested region. The use of this robust trained detector managed to improve the tracking results when running the 2D MOT 2015 sequences as will be seen in the 5.

4.1.2 Tracking

In this part, we illustrate the object model, i.e. the appearance model which uses a separate KCF tracker running for each new target. For detections that do not match any existing KCF trackers, a new tracker would be assigned to this object, and initialized with the

bounding box of this detection. The tracker will build a training model using Gaussian Correlation Filter as in Henriques et al. (2015) to be able to match with further detections in coming frames.

4.1.2.1 Circulant Matrices

KCF trains a linear classifier model using a base sample which is the positive sample and a set of virtual samples which are the negative samples, obtained by a process of translation using a cyclic shift operator as can be seen in Figure 4.2. The signal x becomes identical after n^{th} shifts. The full set of shifted signals is represented as

$$\{P^u x | u = 0, \dots, n-1\} \quad (4.1)$$

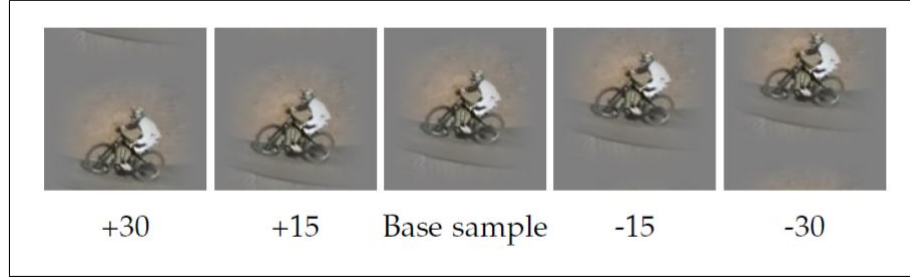
where P is the permutation matrix and n is the number of translation step, and one element of the previous equation set represents a row of a circulant matrix X .

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ x_n & x_1 & x_2 & \dots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \dots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \dots & x_1 \end{bmatrix} \quad (4.2)$$

The use of circulant matrix becomes helpful here, because circular matrices become diagonal in Fourier domain, or in Discrete Fourier Transform (DFT). We can get:

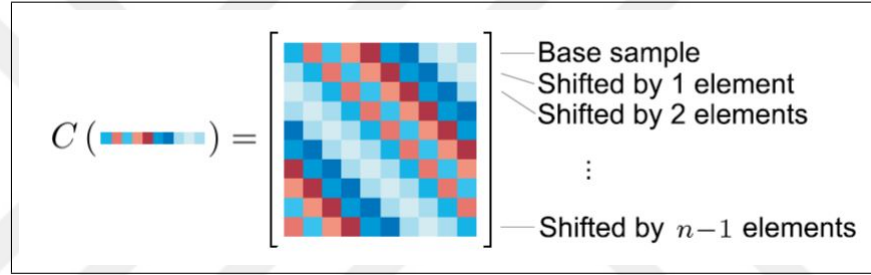
$$X = F \text{diag}(\hat{x}) F^H, \quad (4.3)$$

Figure 4.2: Shifted samples of a base sample that appears in the middle. The translation shift in this figure is done vertically.



Source: Henriques et al. (2015)

Figure 4.3: The circulant matrix illustrated. Each row represents a translated shift of the base sample in 1D.



Source: Henriques et al. (2015)

where F is (DFT) matrix, and $\hat{x}$ is the DFT of the generating vector. Eq. 4.3 expresses the eigen decomposition of a general circulant matrix. The circulant matrix is illustrated in Figure 4.3.

4.1.2.2 Fast Kernel Regression

The kernel matrix $K(n \times n)$ stores the dot-products between all pairs of samples

$$K_{ij} = k(x_i, x_j) = \varphi^T(x_i)\varphi(x_j) \quad (4.4)$$

with high-dimensional space $\varphi(\cdot)$. The following kernels satisfy the condition to claim

that K is circulant Henriques et al. (2015):

- (a) Radial Basic Function kernels -e.g., Gaussian.
- (b) Dot-product kernels - e.g., linear, polynomial.
- (c) Additive kernels -e.g., intersection, x^2 and Hellinger kernels.
- (d) Exponentiated additive kernels. Therefore, the kernelized version of Ridge Regression is possible to diagonal- ize

$$\hat{\alpha}^* = \frac{\hat{y}^*}{\hat{k}^{xx} + \lambda}, \quad (4.5)$$

where $K = C(k^{xx})$ is kernel matrix. Eq. 4.5 is used in the learning phase to update the model for next frame.

4.1.2.3 Initialization

Starting from the size and position information of target in the initial frame, HOG features are extracted from a region that is 2.5 in times of target's size so that the target is highlighted in the center and boundary issues can be avoided. All possible shifts of the target are then calculated from the initial base frame and described by a circulant matrix.

4.1.3 Data Association

Data association is the phase of matching the tracker output to the right object at the corresponding frame and location. In this phase, we have proposed three methods or models to achieve the mission, the three models were all used to calculate the similarity between the trackers output and the previous frame's detections. Later the similarity matrix has been computed using Hungarian algorithm to match a detection to one tracker or object at a time. The models are: Intersection Over Union (IOU), Structural Similarity Index (SSIM), and the mixed model of both methods.

4.1.3.1 Intersection Over Union

In the data association process between detections and the trackers' outputs, the bounding box of each target in the current frame was computed as the output of the KCF tracker that is tracking this target. This tracking output was then compared to the set of detections in this frame through an assignment cost matrix using the intersection-over-union (IOU) distance as a comparison measurement between each detection and the target's assumed bounding box. During the assignment process, the Hungarian algorithm was used to solve the problem optimally. The IOU measure user is calculated as follows:

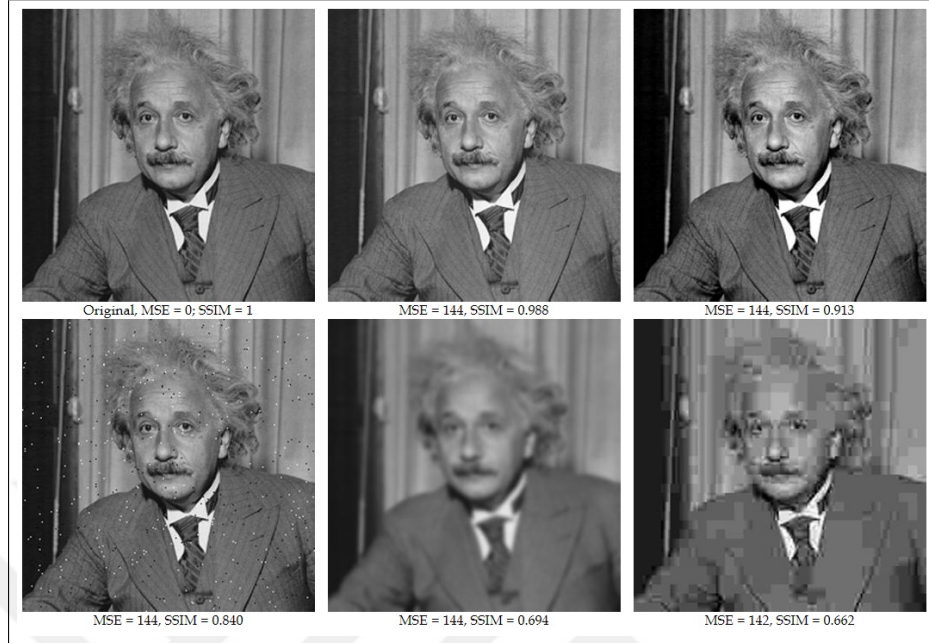
$$IOU(x, y) = \frac{Area(x) \cap Area(y)}{Area(x) \cup Area(y)} \quad (4.6)$$

Equation (4.6) compares each detection in the current frame with the output of KCF tracker at the same frame. The detection with highest IOU would be associated to the matched track if a certain threshold σ_{IOU} . Further, in order to filter false assignments, a minimum threshold of IOU was set to exclude assignments with an overlapping result of less than IOU_{min} .

Regarding the problem of short-term occlusions, the IOU distance comparison that we applied in data association was able to implicitly handle these cases that are caused by passing objects. In this case, when a tracked object is covered by an occluder, only the occluder will be detected and matched with the corresponding detection easily, while the occluded will be detected and matched as long as the visible part of it is tracked by the KCF tracker and matching a corresponding detection. When the occluded gets out of the scene it remains unassigned and can get assigned to the same track if it showed up in a shorter period than a certain threshold.

To improve performance and results the score values of detections were also utilized in the process. Tracks that did not have at least one detection with a score above σ_h were filtered out. This requirement over tracks to include at least one high-scored detection can ensure that the track belongs to a real object of interest, and the other lower-scored

Figure 4.4: SSIM Compared to Mean Squared Error in Image Similarity.



Source: <https://ece.uwaterloo.ca/~z70wang/research/ssim/>

detection can help in completing the track.

4.1.3.2 Structural Similarity Index

Another approach for finding the similarity between a specific detection and a tracking output was to utilize image information again and to compare the similarity between two rectangles based on the image information in each one of them. In this case, we have used the Structural Similarity (SSIM) Index, as implemented in Wang et al. (2004). In this method, a structural similarity quality measure is built from the the perspective of image information. To find the structural information in an image, SSIM separates the influence of the illumination, and defines the the structural information in image as the attributes that represent the structures of objects in the scene, independent of the average values of luminance and contrast. An example is illustrated in Figure 4.4.

SSIM quality assessment index consists of the computation of three terms: the luminance

term, the contrast term and the structural term. The overall index is a multiplicative combination of the three terms.

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (4.7)$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (4.8)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (4.9)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (4.10)$$

where μ_x , μ_y , σ_x , σ_y , and σ_{xy} are the local means, standard deviations, and cross-covariance for images x , y .

To utilize that in our proposed method, we have compared the SSIM between each detection and tracking output and calculated the linear matrix with Hungarian algorithm for each frame. Since both the detection and tracking output are given as a bounding box rectangle, we had to extract the correspondent image rectangle that lies in the bounding box and cropped from the original image. In some cases, the two rectangles did not have the same size, and because SSIM compares between two images that are equal in size we had to resize the rectangle of the tracking output, which may have affected the performance in those cases.

4.1.3.3 Mixed Approach of IOU and SSIM

In this approach, we have applied both function to calculate the similarity matrix. At the first run, IOU was calculated for all trackers and detections, which was not costing because of its speed. Then, SSIM was later applied to compare tracks and detections which already had IOU similarity, and those similarity values were updated with SSIM

comparison values. This ensured that SSIM was applied only when the tracker's output and the detection are in the same region of interest, which means they are not far from each other, and this can eliminate the false cases when SSIM only was used and had to falsely match trackers to some far detections in the scene. This mixed approach, managed to improve the results of SSIM as a data association method. Yet, it has not managed to overperform IOU when used alone which still gave better results. Results and experiments will be later explained in chapter 5.

4.1.3.4 Hungarian Algorithm

The Hungarian Algorithm is a combinational algorithm to solve the assignment problem in polynomial time. The algorithm consists of four steps. At the first run, first and second steps are executed once, and then the third and fourth steps are repeated until an optimal solution for the assignment problem is found. The input of the algorithm is a $n \times n$ matrix. Kuhn (1955).

- (a) Subtract row minima. In this step the lowest element is found for each row and subtracted from element in row.
- (b) Subtract column minima. In this step the lowest element is found for each column and subtracted from each element in that column.
- (c) Cover all zeros with a minimum number of lines. In this step, we need to find the minimum number of lines (horizontal or vertical) that are required to cover all zeros in the resulting matrix. If this minimum number of lines was equal to n , then an optimal assignment solution is found among the zeros, and algorithm stops. If the required minimum number of lines is less than n , then we continue to step 4.
- (d) Create additional zeros. In this step we try to find the smallest element (k) that is not covered by a line in Step 3. k is then subtracted from all uncovered elements and added to all elements that are covered twice.

In our work, we have used the Hungarian Algorithm to solve the assignment matrix along with each one of the data association methods that have been explained earlier.

4.1.4 Creation and Deletion of Track Identities

New identities are going to be created for objects the enter the scene, and those who leave are going to get their identities deleted. In trackers creating, a detection with an overlap less than IOU_{min} is considered as a new track identity. The tracker is initialized with the geometry of the new target's bounding box. A track that is not matched for T_{min} frames is going to be destroyed. T_{min} has been selected among the values 2,3,4 according to the detections set SDP, FRCNN, DPM based on experiments and research done by Bochinski et al. (2017) that is using IOU method for data association.

5. EXPERIMENTAL RESULTS

In this chapter we are going to demonstrate the data that has been used to run, test and evaluate our research. Then, it will further discuss the different experiments that were made in various phases and shows their results. This chapter consists of two sections: Data, and Experiments.

5.1 DATA

The data that has been used in this research was basically both the training and testing datasets of MOT Challenge benchmark Leal-Taixé et al. (2015) and Milan et al. (2016). The MOT Challenge is a benchmark that attempts to establish a unified framework towards important quantification of multi-target tracking. Following this adoption of MOT challenge, our research has followed the benchmark's paradigm of "Tracking by Detection", which uses provided detections (in the form of bounding boxes) in each frame as an input for the tracking method which needs to assign or label each detection to the right object in scene. And then the tracking method would be evaluated through different parameters by the benchmark based on the success in the assigning process between detection and objects at each frame.

5.1.1 Datasets

All datasets are available online on the MOT Challenge website, which include images and detections in text files. Video sequences and their properties are illustrated in Tables 5.1 and 5.2.

Table 5.1: Descriptions of video sequences used (MOT2015 training)

Name	Description
TUD-Stadtmitte	Static camera at about 2m height shows walking people on the street
TUD-Campus	A short sequence with side-view pedestrians
PETS09-S2L1	A widely used sequence showing up to 8 walking pedestrians, partly in unusual patterns
ETH-Bahnhof	Street scene from a moving platform
ETH-Sunnyday	Street scene on a sunny day from a moving platform
ETH-Pedcross2	Street scene from a moving platform
ADL-Rundle-6	A pedestrian street scene filmed from a low angle
ADL-Rundle-8	A pedestrian scene filmed at night by a moving camera
KITTI-13	Busy urban environment filmed from a moving car
KITTI-17	Walking pedestrians on a sunny day, static camera
Venice-2	People walking around a large square

5.1.2 Detections

To run your method on the MOT datasets, you can either adopt and use the public detections that provided by the benchmark or you can use your own detector or any other private detector to generate the detections. In this research, we have used the public detections of MOT15 and MOT17 for training and testing data. In addition, we have also used the detections of Faster Region CNN (FrRCNN) detection framework Ren et al. (2017b) on the training dataset of MOT15, in order to improve the tracking accuracy and compare our results to those of SORT’s tracker Bewley et al. (2016b) in one of our experiments as will be explained in 5. In the public detections of MOT 2015, the detector used was the Fast feature pyramids for object detection Dollár et al. (2014a), which is based on aggregated channel features (ACF). The implementation used the default parameters and the pedestrian model was trained on the INRIA Dalal & Triggs (2005) dataset, rescaled by 0.6 to detect smaller pedestrians.

Table 5.2: Properties of video sequences used

Name	FPS	Tracks	Boxes	Density
TUD-Stadtmitte	25	10	1156	6.5
TUD-Campus	25	8	359	5.1
PETS09-S2L1	7	19	4476	5.6
ETH-Bahnhof	14	171	5415	5.4
ETH-Sunnyday	14	30	1858	5.2
ETH-Pedcross2	14	133	6263	7.5
ADL-Rundle-6	30	24	5009	9.5
ADL-Rundle-8	30	28	6783	10.4
KITTI-13	10	42	762	2.2
KITTI-17	10	9	683	4.7
Venice-2	30	26	7141	11.9

5.1.3 Data Format

The data contains three parts: the images, the detection file, and the annotation files. The images are provided in JPEG format and named sequentially to a 6-digit file name (e.g. 000003.jpg), each set of a specific sequence's images is found in a folder with the sequence name. The detection files are in 'CSV' files, which are provided as one detection file per sequence. The file contains the coordinates of each moving object, represented by a bounding box with the coordinates of the top-left corner (x,y) as well as the width and height of the bounding box, in the stated order. In the file, each line represents one object instance, and contains 9 values as shown in Figure 5.1 for MOT16, and MOT17 benchmarks, and Figure 5.2 for MOT15.

The difference between the data formats of the two the benchmarks can be seen only after the 7th value. Where in Figure 5.1 MOT16/17, the 8th and 9th values refer to class of object annotated, and object's visibility ratio (between 0 and 1), the values from 8 to 10 in MOT15 in Figure 5.2 refer the (x,y,z) coordinates of the 3D position of the object, which is not applicable in the 2D datasets (the target area of this research) and has the value of -1 instead.

Figure 5.1: Data format for both detection files and annotation/ground truth files in MOT16/MOT17 benchmarks

Position	Name	Description
1	Frame number	Indicate at which frame the object is present
2	Identity number	Each pedestrian trajectory is identified by a unique ID (−1 for detections)
3	Bounding box left	Coordinate of the top-left corner of the pedestrian bounding box
4	Bounding box top	Coordinate of the top-left corner of the pedestrian bounding box
5	Bounding box width	Width in pixels of the pedestrian bounding box
6	Bounding box height	Height in pixels of the pedestrian bounding box
7	Confidence score	DET: Indicates how confident the detector is that this instance is a pedestrian. GT: It acts as a flag whether the entry is to be considered (1) or ignored (0).
8	Class	GT: Indicates the type of object annotated
9	Visibility	GT: Visibility ratio, a number between 0 and 1 that says how much of that object is visible. Can be due to occlusion and due to image border cropping.

Source: Milan et al. (2016)

Figure 5.2: Data format for both detection files and annotation/ground truth files in MOT15 benchmarks

Position	Name	Description
1	Frame number	Indicate at which frame the object is present
2	Identity number	Each pedestrian trajectory is identified by a unique ID (−1 for detections)
3	Bounding box left	Coordinate of the top-left corner of the pedestrian bounding box
4	Bounding box top	Coordinate of the top-left corner of the pedestrian bounding box
5	Bounding box width	Width in pixels of the pedestrian bounding box
6	Bounding box height	Height in pixels of the pedestrian bounding box
7	Confidence score	Indicates how confident the detector is that this instance is a pedestrian. For the ground truth and results, it acts as a flag whether the entry is to be considered.
8	x	3D x position of the pedestrian in real-world coordinates (−1 if not available)
9	y	3D y position of the pedestrian in real-world coordinates (−1 if not available)
10	z	3D z position of the pedestrian in real-world coordinates (−1 if not available)

Source: Leal-Taixé et al. (2015)

The 7th value, which indicates the "confidence score" of the detector that this is object is a pedestrian, has been utilized in this research in the process of accepting the detection before matching it with the tracker's output, the threshold value has been varied based on the dataset and sequence as will be illustrated in the next section 4.1.

The annotation or ground truth (GT) files follow the same structure explained above. The difference would be in the second value (Identity number). Where in the detection files this values is always set to (-1), in the ground truth or output files it holds the number of the object that this instance belongs to. In the evaluation process, this value in output file is compared against the corresponding value in the ground truth in the same line.

5.2 EXPERIMENTS

The performance of this research implementation has been evaluated by running on the training sequences of MOT17 benchmark database Milan et al. (2016). The benchmark includes various test cases such moving and static camera recordings. For tuning IOU_{min} , and T_{min} parameters, the same training/validation split was reported as in Bochinski et al. (2017). In our experiments, we have compared our results to SORT Bewley et al. (2016a) one of the available algorithms for testing on the same data set, and that follows the same paradigm of tracking-by-detection, but follows the motion model when processing images instead of the appearance that we are adopting. In Bewley et al. (2016a) they have used Kalman Filter for tracking as opposed to our KCF tracker. Thus, the results can be also used to compare between an appearance-model-adopting method and a motion-model-adopting one.

Figure 5.3: Labeled objects as seen from the output frames of our method on the DPM dataset.

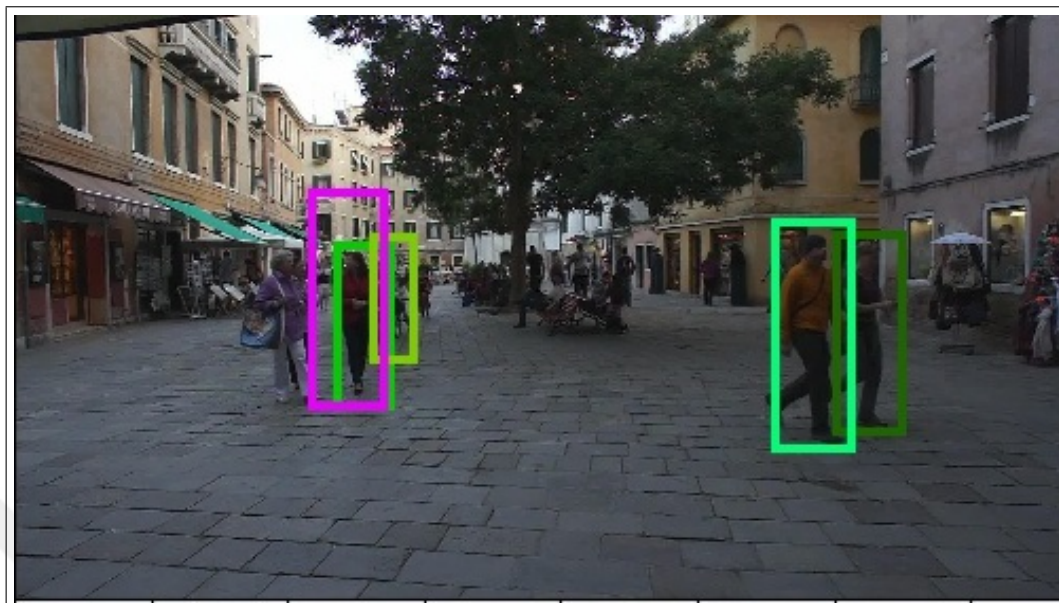


Figure 5.4: Labeled objects as seen from the output frames of our method on the DPM dataset.

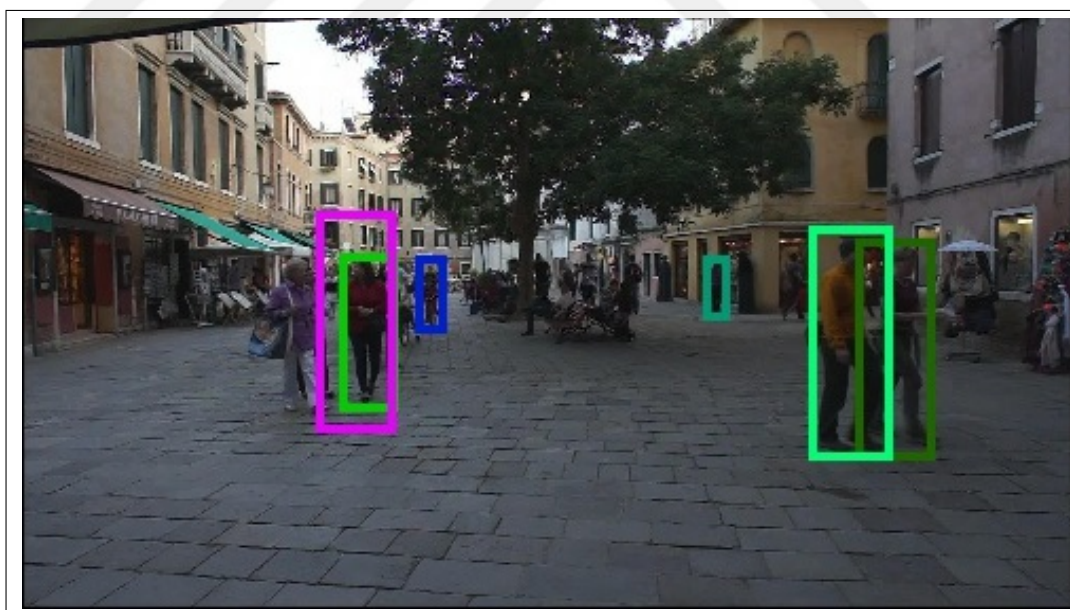
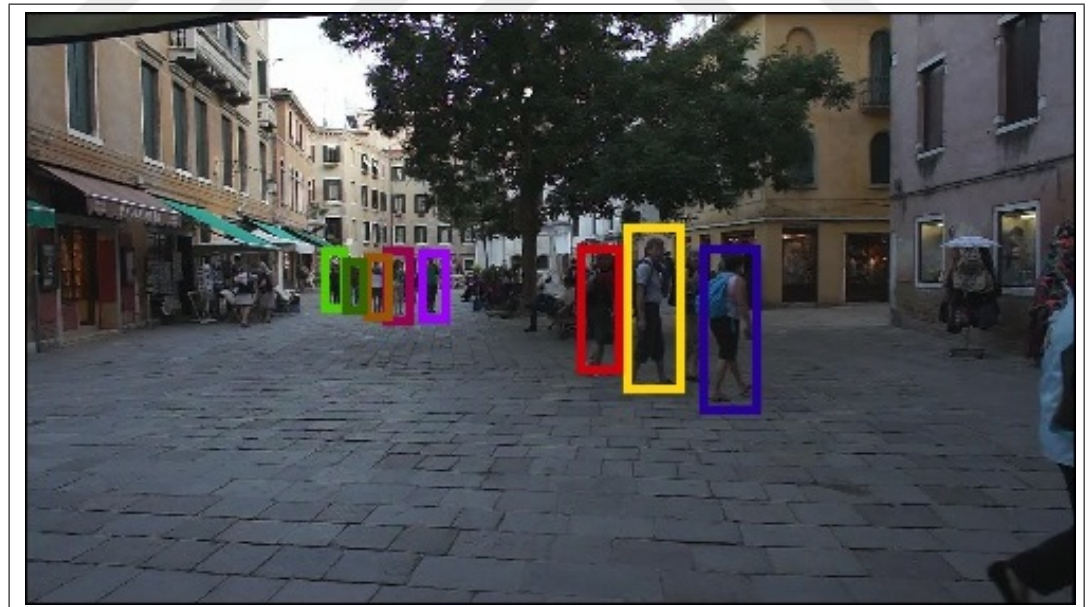


Figure 5.5: Labeled objects as seen from the output frames of our method on the DPM dataset



Figure 5.6: Labeled objects as seen from the output frames of our method on the DPM dataset



In addition, we have run our code on the test sequences of MOT15 by Leal-Taixé et al. (2015) and submitted our results on the benchmark, which were accepted and published

and ranked as 73rd among all submitted algorithms in terms of MOTA, which is explained in section 5.2.1. The results can be seen in figure 5.7. Labeled samples from running our method on the video sequence can be seen in Figures 5.3, 5.4, 5.5 and 5.6.

5.2.1 Metrics

The metrics of MOT benchmark Milan et al. (2016) have been used to evaluate multiple object tracking performance, along with some other metrics:

- (a) $MOTA(\uparrow)$: Multiple Object Tracking Accuracy. It is the most widely used metric to evaluate tracker's performance. It combines three error resources as follows:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \quad (5.1)$$

Leal-Taixé et al. (2015)

- (b) $MOTP(\uparrow)$: Multiple Object Tracking Precision. Represents the average dissimilarity between all true positives and the ground truth targets that are corresponding to them. For a bounding box over-lap, it is given as:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \quad (5.2)$$

where c_t is the number of matches in frame t and $d_{t,i}$ is the bounding box overlap of object i with its assign ground truth object. MOTP gives the average overlap between hypotheses and their respective objects. It ranges between $t_d := 50\%$ and 100% . Leal-Taixé et al. (2015).

- (c) $FAF(\downarrow)$: a metric that counts false alarms per frame.
- (d) $MT(\uparrow)$: a metric that counts mostly tracked trajectories. This means the number of targets that have the same label for at least 80% of their life span. It is irrelevant for

this metric whether the identity remains the same throughout the track. A higher number of MT is desired.

- (e) ML($\downarrow$): number of mostly lost trajectories. This means the number of targets that have the same label for at least 20% of their life span.
- (f) FM($\downarrow$): number of track fragmentations. Counts how many times a ground truth trajectory is interrupted (untracked) Leal-Taixé et al. (2015).
- (g) IDs: ID switch ratio.
- (h) TP: number of true positive assignments.
- (i) FP($\downarrow$): a metric that counts false detections.
- (j) FN($\downarrow$): a metric that counts missed detections.
- (k) Frag($\downarrow$): a metric that counts fragmentations in the case of a miss detection interrupts a track.

5.2.2 Performance Evaluation

The types of experiments that we have run and evaluated were as follows:

5.2.2.1 Comparing Proposed Approach Against Other Data Association Methods

Table 5.3: Performance results of the proposed method comparing three methods for data association

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	ID_S $\downarrow$
IOU (Proposed)	23.4	20.6	71.9	74	219	9867	20698	1135
SSIM	17.0	-2.0	72.0	51	200	11928	21170	7623
Mixed	21.7	14.7	72.0	45	220	9239	21990	2805

Table 5.5 shows the results of our method using the SSIM comparison in data association replacing (IOU only). From these results, we notice that the overall accuracy was not improved and the IOU comparison in all frames had better overall accuracy and performance.

Table 5.4: Detailed performance of all sequences in the MOT2015 training data set of the proposed approach using IOU

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML	FP	FN	FM	ID_S
TUD-Stadtmitte	54.5	48.4	65.3	4	0	168	362	54	66
TUD-Campus	44.8	40.1	71.0	1	0	44	153	37	18
PETS09-S2L1	67.2	62.0	71.1	14	0	804	662	269	234
ETH-Bahnhof	26.5	23.0	73.1	43	59	1924	2052	356	191
ETH-Sunnyday	33.1	31.3	75.8	2	16	109	1133	100	35
ETH-Pedcross2	8.5	7.8	71.1	1	113	198	5529	109	50
ADL-Rundle-6	23.4	19.4	72.0	0	2	1320	2516	287	200
ADL-Rundle-8	5.4	3.6	72.0	4	8	2452	3964	314	123
KITTI-13	16.9	12.1	71.2	1	15	207	425	35	38
KITTI-17	50.5	47.9	70.8	0	0	50	287	38	19
Venice-2	13.1	10.8	72.7	4	6	2591	3615	266	161

In this experiment we have compared the KCF approach using three models of data association that we have proposed and explained in 4. These were: IOU, SSIM, and a mixed approach. By adopting one different method at a time for data association along with KCF as a tracking method, it can be shown from the results that the best overall accuracy was achieved through the full IOU approach.

Table 5.5: Performance results of the proposed method comparing three methods for data association

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	ID_S $\downarrow$
IOU (Proposed)	23.4	20.6	71.9	74	219	9867	20698	1135
SSIM	17.0	-2.0	72.0	51	200	11928	21170	7623
Mixed	21.7	14.7	72.0	45	220	9239	21990	2805

Table 5.5 shows the results of our method using the SSIM comparison in data association replacing (IOU only). From these results, we notice that the overall accuracy was not

improved and the IOU comparison in all frames had better overall accuracy and performance.

Table 5.6: Detailed performance of all sequences in the MOT2015 training data set of the proposed approach using IOU

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML	FP	FN	FM	ID_S
TUD-Stadtmitte	54.5	48.4	65.3	4	0	168	362	54	66
TUD-Campus	44.8	40.1	71.0	1	0	44	153	37	18
PETS09-S2L1	67.2	62.0	71.1	14	0	804	662	269	234
ETH-Bahnhof	26.5	23.0	73.1	43	59	1924	2052	356	191
ETH-Sunnyday	33.1	31.3	75.8	2	16	109	1133	100	35
ETH-Pedcross2	8.5	7.8	71.1	1	113	198	5529	109	50
ADL-Rundle-6	23.4	19.4	72.0	0	2	1320	2516	287	200
ADL-Rundle-8	5.4	3.6	72.0	4	8	2452	3964	314	123
KITTI-13	16.9	12.1	71.2	1	15	207	425	35	38
KITTI-17	50.5	47.9	70.8	0	0	50	287	38	19
Venice-2	13.1	10.8	72.7	4	6	2591	3615	266	161

Table 5.7: Detailed performance of all sequences in the MOT2015 training set of the proposed approach using SSIM only

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML	FP	FN	FM	ID_S
TUD-Stadtmitte	45.8	28.0	65.3	4	0	224	400	80	208
TUD-Campus	30.9	2.5	71.7	0	0	67	179	48	104
PETS09-S2L1	58.7	29.8	71.3	11	0	999	847	406	1294
ETH-Bahnhof	18.4	-5.5	72.9	29	53	2185	2231	553	1298
ETH-Sunnyday	27.7	9.2	76.0	0	12	131	1210	199	346
ETH-Pedcross2	6.5	1.1	71.2	0	110	268	5585	205	338
ADL-Rundle-6	15.7	-6.6	71.7	0	2	1695	2524	449	1119
ADL-Rundle-8	-1.8	-20.7	72.1	3	8	2938	3961	539	1287
KITTI-13	-1.3	-23.6	70.8	0	11	345	425	65	172
KITTI-17	40.5	17.3	71.0	0	0	82	322	66	161
Venice-2	9.3	-9.6	72.7	4	4	2958	3515	449	1351

However, by reviewing both tables 5.6 and 5.8, we can notice that although IOU is performing better in the overall results, the mixed model of (IOU and SSIM) is out performing in the video sequence of (ADL-Rundle-8), which differs from other sequences by the fact that it is a night scene. We can notice here that the fact that SSIM is utilizing the illumination effect of the image in measuring similarity has managed to pay off in processing

Table 5.8: Detailed performance of all sequences in the MOT2015 training set of the proposed approach using the mixed IOU&SSIM

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML	FP	FN	FM	ID_s
TUD-Stadtmitte	48.5	39.3	65.3	4	0	185	408	76	109
TUD-Campus	41.3	29.2	71.3	1	0	45	164	39	45
PETS09-S2L1	62.0	52.7	71.2	8	0	634	1066	394	418
ETH-Bahnhof	24.8	17.5	73.1	19	63	1577	2491	479	399
ETH-Sunnyday	26.8	20.1	76.4	0	18	86	1272	130	126
ETH-Pedcross2	7.1	4.7	71.1	0	112	197	5261	160	150
ADL-Rundle-6	27.9	11.1	71.5	0	3	1232	2678	424	541
ADL-Rundle-8	8.7	4.6	72.3	3	8	2100	4090	397	281
KITTI-13	12.4	4.1	71.4	1	17	177	489	49	65
KITTI-17	39.6	33.7	71.3	0	2	59	352	49	42
Venice-2	13.1	6.2	72.2	3	6	2473	3728	407	498

night scenarios. As a conclusion, we can say that the mixed model of (IOU and SSIM) is better a choice in night scenarios than the IOU only approach.

5.2.2.2 Comparing Proposed Approach Against Trackers with Same Data Association Method

In evaluating tracking performance, the MOT benchmark Leal-Taixé et al. (2015) tests were used, which compare the results to the ground truth for 11 sequences. In table 5.9, the results of our proposed method were compared along with other trackers, such as SORT Bewley et al. (2016a) which was previously mentioned. In table 5.9, we can see that our proposed method precedes that of SORT with the significant parameters MOTAL, MOTA which represent the total accuracy of tracking in the overall videos of benchmark.

In table 5.10, we can also see the comparison of our results against on SORT algorithm and baseline methods on the 2D MOT 2015. Table 5.10 shows that our method precedes that of SORT in accuracy by more than 4 points as measured in MOTA and MOTAL.

Table 5.9: Performance results of the proposed method on MOT17 training data sets and public detections

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	ID_S $\downarrow$
(Proposed)	46.3	45.4	83.9	253	657	6625	174401	3031
SORT	45.4	44.6	84.7	267	652	14921	169004	2743

Table 5.10: Performance results of the proposed method on MOT15 training data set and FrCNN private detections

Method	MOTAL $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML $\downarrow$	FP $\downarrow$	FN $\downarrow$	ID_S $\downarrow$
(Proposed)	38.2	36.5	71.8	95	161	6182	18491	661
SORT	34.0	32.9	73.7	112	164	5386	20930	442

5.2.2.3 Comparing Proposed KCF Approach Against Other Trackers on Benchmark

In this part, we demonstrate our results running on the testing data sequences of MOT 2015. Our method has been accepted and published on the MOT Challenge website. In Figure 5.7, we can see our results in comparison to other trackers submitted on the benchmark.

Figure 5.7: Published results of trackers submitted online on MOT15 benchmark.

Multiple Object Tracking Benchmark												
home	data	results	vis	QVA	submit	FAQ	people	login	sign up			
Tracker	Avg Rank	↑MOTA	IDF1	MT	ML	FP	FN	ID Sw.	Frag	Hz	Detector	
TSDA_OAL	41.1	18.6 ±17.6	36.1	9.4%	42.3%	16,350	32,853	806 (17.3)	1,544 (33.2)	19.7	Public	
71.	H. Ko. Online multi-person tracking with two-stage data association and online appearance model learning. In IET Computer Vision, 2017.											
GMPHD	41.3	18.5 ±12.7	28.4	3.9%	55.3%	7,864	41,766	459 (14.3)	1,266 (39.5)	19.8	Public	
72.	Y. Song, M. Jeon. Online Multiple Object Tracking with the Hierarchically Adopted GM-PHD Filter using Motion and Appearance. In IEEE/IEE The International Conference on Consumer Electronics (ICCE) Asia, 2016.											
KCF_Simple	50.3	18.3 ±11.1	25.1	2.6%	49.8%	8,976	39,805	1,436 (40.8)	2,634 (74.8)	35.6	Public	
73.	Anonymous submission											
SMOT	57.3	18.2 ±10.3	0.0	2.8%	54.8%	8,780	40,310	1,148 (33.4)	2,132 (62.0)	2.7	Public	
74.	C. Dicle, O. Camps, M. Sznaler. The Way They Move: Tracking Targets with Similar Appearance. In ICCV, 2013.											
Bar_dist	41.5	18.0 ±16.0	17.6	5.7%	35.9%	10,019	36,222	4,161 (101.4)	5,043 (122.9)	2,891.5	Public	
75.	Anonymous submission											
ALEXTRAC	50.8	17.0 ±12.1	17.3	3.9%	52.4%	9,233	39,933	1,859 (53.1)	1,872 (53.5)	3.7	Public	
76.	A. Bewley, L. Ott, F. Ramos, B. Upcroft. ALETRAC: Affinity Learning by Exploring Temporal Reinforcement within Association Chains. In International Conference on Robotics and Automation (ICRA), (to appear) 2016.											
TBD	56.0	15.9 ±17.6	0.0	6.4%	47.9%	14,943	34,777	1,939 (44.7)	1,963 (45.2)	0.7	Public	
77.	A. Geiger, M. Lauer, C. Wojek, C. Stiller, R. Urtasun. 3D Traffic Scene Understanding from Movable Platforms. In Pattern Analysis and Machine Intelligence (PAMI), 2014.											
GSCR	42.0	15.8 ±10.5	27.9	1.8%	61.0%	7,597	43,633	514 (17.7)	1,010 (34.8)	28.1	Public	
78.	L. Fagot-Bouquet, R. Audigier, Y. Dhome, F. Lerasle. Online multi-person tracking based on global sparse collaborative representations. In ICIP, 2015.											
TC_ODAL	56.7	15.1 ±15.0	0.0	3.2%	55.8%	12,970	38,538	637 (17.1)	1,716 (46.0)	1.7	Public	
79.	S. Bae, K. Yoon. Robust Online Multi-Object Tracking based on Tracklet Confidence and Online Discriminative Appearance Learning. In CVPR, 2014.											
EFMC	49.2	14.9 ±9.8	11.5	3.3%	50.5%	5,583	41,085	5,623 (169.7)	3,443 (103.9)	24.6	Public	
80.	Anonymous submission											
Tracker	Avg Rank	↑MOTA	IDF1	MT	ML	FP	FN	ID Sw.	Frag	Hz	Detector	
DP_NMS	44.8	14.5 ±14.5	19.7	6.0%	40.8%	13,171	34,814	4,537 (104.7)	3,090 (71.3)	444.8	Public	

Figure 5.8: Published results of some other trackers on MOT15 benchmark.

Multiple Object Tracking Benchmark												
home	data	results	vis	QVA	submit	FAQ	people	login	sign up			
Tracker	Avg Rank	↑MOTA	IDF1	MT	ML	FP	FN	ID Sw.	Frag	Hz	Detector	
DP_NMS	44.8	14.5 ±14.5	19.7	6.0%	40.8%	13,171	34,814	4,537 (104.7)	3,090 (71.3)	444.8	Public	
81.	H. Pirsiavash, D. Ramanan, C. Fowlkes. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In CVPR, 2011.											
LDCT	41.5	4.7 ±41.3	16.8	11.4%	32.5%	14,066	32,156	12,348 (259.1)	2,918 (61.2)	20.7	Public	
82.	F. Solera, S. Calderara, R. Cucchiara. Learning to Divide and Conquer for Online Multi-Target Tracking. In ICCV, 2015.											
JPDA_OP	38.3	3.6 ±11.3	7.5	0.4%	96.1%	1,024	58,189	29 (5.5)	119 (22.5)	77.7	Public	
83.	Anonymous submission											
NEW_NW	50.1	-38.8 ±53.1	2.0	0.1%	98.3%	25,606	59,569	80 (26.3)	112 (36.8)	5.8	Public	
84.	Anonymous submission											

In general, we notice from 5.6 and 5.1 that our method was able to give the better results in static camera scenes such as PETS09-S2L1, TUD-Stadtmitte and KITTI-17, in com-

parison to scenes with a moving camera, such as: ETH-Bahnhof, ETH-Pedcross2 and ADL-Rundle-8. Which shows that the algorithm may lose some of its accuracy when applied to scenarios of the later type.

We can also notice that when the boxes in the sequence was highly increasing, like those with above 4000 boxes per sequence, the overall accuracy of was decreasing. This can be due to the increasing in similarities between the several boxes which makes the matching and assignment problem harder.

In the sequence ADL-Rundle-8, which is recorded in a night scenario, we notice that the mixed model was outperforming the IOU model in terms of accuracy and the number of tracked objects. We can observe here that the utilization of SSIM to discriminate between close objects was able to solve a problem in night scenes, due to the calculation of the luminance and contrast similarities in the process of data associating tracks to corresponding detections.

Overall, IOU as a data association method was able to perform better in most of the tested sequences. We refer this to that the use of IOU was able to give more confidence to the tracking results of KCF without further changes in terms of image information, and thus adopting a robust method in handling image similarities which can perform better than other simpler image similarity handling functions.

6. CONCLUSIONS

This chapter reviews our conclusions that have been made during this research in addition to discussing the possibilities of future work.

In this research, we have introduced a solution that addressed the MOT problem, and pedestrians particularly. In addition, our solution was generic that can also operate to track other type of objects by choosing the relevant detector as input. In our work, we have combined the features of Kernelized Correlation Filters in tracking along with other methods for data association such Intersection Over Union and Hungarian algorithm. The utilizing of a robust and fast tracker like KCF allowed us to achieve higher accuracy than the state-of-art methods such as MOT Benchmark's baseline trackers, and other approaches that adopted the same data association methods as us.

In our experiments, we have proved that KCF and appearance model perform better in multiple object tracking than motion model trackers such as Kalman Filter. Moreover, by running our approach using several methods for data association, we found that, despite its simplicity, intersection over union and Hungarian algorithm formed a suitable way to tackle the data association problem of tracking-by-detection, and achieved more accurate comparisons between trackers and detections than Structural Similarity Index method that uses the image information.

We have also found that the robustness of the detector would directly affect and enhance the final tracking results. In our experiments, by replacing the benchmark's public detections with the detections generated by a trained FrCNN detector, the tracking were results were improved by a great factor.

Generally speaking, we concluded that our method was able to give the better results in static camera scenes in comparison to scenes with a moving camera, which shows that the algorithm may lose some of its accuracy when applied to scenarios of the later type. We can also notice that when the boxes in the sequence was highly increasing, the overall

accuracy of was decreasing. This can be due to the increasing in similarities between the several boxes which makes the matching and assignment problem harder.

From night scene sequences, we noticed that the mixed model was outperforming the IOU model in terms of accuracy and the number of tracked objects. We can observe here that the utilization of SSIM to discriminate between close objects was able to solve a problem in night scenes, due to the calculation of the luminance and contrast similarities in the process of data associating tracks to corresponding detections.

Overall, IOU as a data association method was able to perform better in most of the tested sequences. We refer this to that the use of IOU was able to give more confidence to the tracking results of KCF without further changes in terms of image information, and thus adopting a robust method in handling image similarities which can perform better than other simpler image similarity handling functions.

In real life applications, the work done in this research can be used as a basis for video surveillance softwares. By adapting the inputs into different types of trained detectors, the work can also be applied on tracking other types of objects, such as cars, bicycles or animals.

In future work, this research can be extended to to address Human Computer Interface (HCI) problems by using the tracking output results as a basis for extraction of visual information such as expressions and gestures in order to achieve advanced HCI. In addition, it can also be a a basis for applications in medical image processing to label multiple cells in images, or in virtual augmented reality to supply users with tracking experience in video conferences.

In addition, as we have used the classical version of KCF in Henriques et al. (2015), other versions can also be applied to achieve this work. This can include replacing the HOG feature extraction with hand-crafted features (fHoG + color-naming) or deep convolutional features as in Uzkent & Seo (2018).

REFERENCES

Periodicals

- Andriyenko, A. & Schindler, K., 2011. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, pp. 1265–1272.
- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B., 2016a. Simple online and real-time tracking. In *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, pp. 3464–3468.
- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B., 2016b. Simple online and realtime tracking. In *Proceedings - International Conference on Image Processing, ICIP*.
- Bewley, A., Ott, L., Ramos, F., & Upcroft, B., 2016c. Alextrac: Affinity learning by exploring temporal reinforcement within association chains. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 2212–2218.
- Bochinski, E., Eiselein, V., & Sikora, T., 2017. High-speed tracking-by-detection without using image information. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, pp. 1–6.
- Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M., 2010. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 2544–2550.
- Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., & Van Gool, L., 2009. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 1515–1522.
- Dalal, N. & Triggs, B., 2005. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* **1**, pp. 886–893 vol. 1.
- Danelljan, M., Häger, G., Khan, F., & Felsberg, M., 2014. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press.
- Dicle, C., Camps, O. I., & Sznaiier, M., 2013. The Way They Move: Tracking Multiple Targets with Similar Appearance. In *2013 IEEE International Conference on Computer Vision*. pp. 2304–2311.
- Dollár, P., Appel, R., Belongie, S., & Perona, P., 2014a. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36(8)**, pp. 1532–1545.

- Dollár, P., Appel, R., Belongie, S., & Perona, P., 2014b. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36(8)**, pp. 1532–1545.
- Ess, A., Leibe, B., Schindler, K., & Van Gool, L., 2009. Robust multiperson tracking from a mobile platform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31(10)**, pp. 1831–1846.
- Gavrila, D. M. & Munder, S., 2007. Multi-cue pedestrian detection and tracking from a moving vehicle. *International journal of computer vision* **73(1)**, pp. 41–59.
- Henriques, J. F., Caseiro, R., Martins, P., & Batista, J., 2015. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Kim, C., Li, F., Ciptadi, A., & Rehg, J. M., 2015. Multiple Hypothesis Tracking Revisited. In *2015 IEEE International Conference on Computer Vision (ICCV)*. pp. 4696–4704.
- Kratz, L. & Nishino, K., 2010a. Tracking with local spatio-temporal motion patterns in extremely crowded scenes .
- Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Čehovin, L., Nebehay, G., Vojir, T., Fernandez, G., Lukežič, A., Dimitriev, A. et al., 2014. The visual object tracking vot2014 challenge results,” 2014.
- Kuhn, H. W., 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly* **2(1-2)**, pp. 83–97.
- Kuo, C.-H., Huang, C., & Nevatia, R., 2010. Multi-target tracking by on-line learned discriminative appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 685–692.
- Leal-Taixé, L., Milan, A., Reid, I., Roth, S., & Schindler, K., 2015. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking .
- Leibe, B., Schindler, K., Cornelis, N., & Van Gool, L., 2008. Coupled object detection and tracking from static cameras and moving vehicles. *IEEE transactions on pattern analysis and machine intelligence* **30(10)**, pp. 1683–1698.
- Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **60(2)**, pp. 91–110.
- Lu, W.-L., Ting, J.-A., Little, J. J., & Murphy, K. P., 2013. Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35(7)**, p. 1704–1716.
- Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., & Kim, T.-K., 2014. Multiple object tracking: A literature review. *arXiv preprint arXiv:1409.7618* .

- Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K., 2016. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831* .
- Milan, A., Roth, S., & Schindler, K., 2014. Continuous energy minimization for multitarget tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(1), pp. 58–72.
- Mitzel, D., Horbert, E., Ess, A., & Leibe, B., 2010. Multi-person tracking with sparse detection and continuous segmentation. In *European Conference on Computer Vision*. Springer, pp. 397–410.
- Porikli, F., Tuzel, O., & Meer, P., 2006. Covariance tracking using model update based on lie algebra. In *null*. IEEE, pp. 728–735.
- Ren, S., He, K., Girshick, R., & Sun, J., 2017b. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Ren, S., He, K., Girshick, R. B., & Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, pp. 1137–1149.
- Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M., 2014. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
- Sugimura, D., Kitani, K. M., Okabe, T., Sato, Y., & Sugimoto, A., 2009. Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait. In *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 1467–1474.
- Uzkent, B. & Seo, Y., 2018. Enkcf: Ensemble of kernelized correlation filters for high-speed object tracking. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* , pp. 1133–1141.
- Wang, Z., Bovik, A., Rahim Sheikh, H., & Simoncelli, E., 2004. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on* **13**, pp. 600 – 612.
- Wu, Z., Thangali, A., Sclaroff, S., & Betke, M., 2012. Coupling detection and data association for multiple object tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1948–1955.
- Xing, J., Ai, H., & Lao, S., 2009. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 1200–1207.

- Xing, J., Ai, H., Liu, L., & Lao, S., 2011. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *IEEE Transactions on Image Processing* **20(6)**, pp. 1652–1667.
- Yamaguchi, K., Berg, A. C., Ortiz, L. E., & Berg, T. L., 2011. Who are you with and where are you going? In *CVPR 2011*. pp. 1345–1352.
- Yang, B. & Nevatia, R., 2012. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 2034–2041.
- Yang, F., Choi, W., & Lin, Y., 2016. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2129–2137.
- Yoon, J. H., Yang, M., Lim, J., & Yoon, K., 2015. Bayesian Multi-object Tracking Using Motion Context from Multiple Objects. In *2015 IEEE Winter Conference on Applications of Computer Vision*. pp. 33–40.
- Zhao, X., Gong, D., & Medioni, G., 2012. *Tracking using motion patterns for very crowded scenes*. Springer.