Titre: Title:	Robust Face Tracking in Video Sequences
Auteur: Author:	Tanushri Chakravorty
Date:	2017
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Chakravorty, T. (2017). Robust Face Tracking in Video Sequences [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2768/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2768/
Directeurs de recherche: Advisors:	Guillaume-Alexandre Bilodeau, & Éric Granger
Programme: Program:	Génie informatique

UNIVERSITÉ DE MONTRÉAL

ROBUST FACE TRACKING IN VIDEO SEQUENCES

TANUSHRI CHAKRAVORTY
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
SEPTEMBRE 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ROBUST FACE TRACKING IN VIDEO SEQUENCES

présentée par : CHAKRAVORTY Tanushri
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. GAGNON Michel, Ph. D., président
M. BILODEAU Guillaume-Alexandre, Ph. D., membre et directeur de recherche
M. GRANGER Éric, Ph. D., membre et codirecteur de recherche
M. ALOISE Daniel, Ph. D., membre
M. CLARK James J., P. Eng., membre externe

DEDICATION

“The most exciting phrase to hear in science,
the one that heralds new discoveries, is
not ’Eureka!’ but ’That’s funny...’ ”

- Isaac Asimov

Dedicated to the curious minds......

ACKNOWLEDGEMENTS

I would first like to express my sincere gratitude to my research advisers, Prof. Guillaume-Alexandre Bilodeau and Prof. Eric Granger, for giving me this opportunity to research under their guidance. Their sincere efforts and discussions have helped me to surpass the challenges during this project.

I am also thankful to Natural Sciences and Engineering Research Council of Canada (NSERC), FQRNT Québec and REPARTI, for providing funds to carry out this research.

I would also like to thank my colleagues at LITIV and LIVIA lab. I express my gratitude to my friends for being there throughout this journey.

I would like to thank my parents for their guidance, support, love and care. I would also like to thank my sister, as her work has always inspired me to think beyond the fuzziness of technology, art and culture.

A huge thanks to the amazing musicians for their inspiring music and to my amazing Sennheiser headphones.

Finally, I would like to thank members of the jury for accepting to review the thesis and for their valuable comments.

RÉSUMÉ

Ce travail présente une analyse et une discussion détaillées d'un nouveau système de suivi des visages qui utilise plusieurs modèles d'apparence ainsi qu'un e approche suivi par détection. Ce système peut aider un système de reconnaissance de visages basé sur la vidéo en donnant des emplacements de visages d'individus spécifiques (région d'intérêt, ROI) pour chaque cadre. Un système de reconnaissance faciale peut utiliser les ROI fournis par le suivi du visage pour obtenir des preuves accumulées de la présence d'une personne d'une personne présente dans une vidéo, afin d'identifier une personne d'intérêt déjà inscrite dans le système de reconnaissance faciale.

La tâche principale d'une méthode de suivi est de trouver l'emplacement d'un visage présent dans une image en utilisant des informations de localisation à partir de la trame précédente. Le processus de recherche se fait en trouvant la meilleure région qui maximise la possibilité d'un visage présent dans la trame en comparant la région avec un modèle d'apparence du visage. Cependant, au cours de ce processus, plusieurs facteurs externes nuisent aux performances d'une méthode de suivi. Ces facteurs externes sont qualifiés de nuisances et apparaissent habituellement sous la forme d'une variation d'éclairage, d'un encombrement de la scène, d'un flou de mouvement, d'une occlusion partielle, etc. Ainsi, le principal défi pour une méthode de suivi est de trouver la meilleure région malgré les changements d'apparence fréquents du visage pendant le processus de suivi. Étant donné qu'il n'est pas possible de contrôler ces nuisances, des modèles d'apparence faciale robustes sont conçus et développés de telle sorte qu'ils soient moins affectés par ces nuisances et peuvent encore suivre un visage avec succès lors de ces scénarios.

Bien qu'un modèle d'apparence unique puisse être utilisé pour le suivi d'un visage, il ne peut pas s'attaquer à toutes les nuisances de suivi. Par conséquent, la méthode proposée utilise plusieurs modèles d'apparence faciale pour s'attaquer à ces nuisances. En outre, la méthode proposée combine la méthodologie *du suivi par détection* en employant un détecteur de visage qui fournit des rectangles englobants pour chaque image. Par conséquent, le détecteur de visage aide la méthode de suivi à aborder les nuisances de suivi. De plus, un détecteur de visage contribue à la réinitialisation du suivi pendant un cas de dérive. Cependant, la précision suivi peut encore être améliorée en générant des candidats additionels autour de l'estimation de la position de l'objet par la méthode de suivi et en choisissant le meilleur parmi eux. Ainsi, dans la méthode proposée, le suivi du visage est formulé comme le visage candidat qui maximise la similitude de tous les modèles d'apparence.

Bien que tous les mécanismes de suivi susmentionnés soient utilisés pendant le processus de suivi des visages, il est essentiel de mettre à jour les modèles d'apparence pour faire face aux changements d'apparence de celui-ci. Une mise à jour en temps opportun agit comme une mémoire temporelle qui explique l'apparence des changements. Par conséquent, cette mise à jour du modèle nécessite une stratégie qui doit être prudente, sinon une mise à jour intempestive entraînera une contamination du modèle. Dans notre méthode proposée, nous effectuons deux types de mises à jour : mise à jour de l'apparence à long terme et mise à jour de l'apparrence à court terme. La mise à jour de l'apparence à long terme conserve les caractéristiques d'apparence fiables qui sont utiles pour une localisation précise du visage. Par conséquent, ces mises à jour se produisent pendant toute la séquence vidéo. D'autre part, la mise à jour de l'apparence à court terme permet de suivre le visage lors d'un brusque changement d'apparence dû aux nuisances de suivi. Cela permet d'éviter les éventuelles défaillances de suivi telles que la dérive. Les mises à jour à court terme sont régies par une stratégie de détection d'occlusion et en analysant la précision d'une caractéristique pour la localisation du visage.

Enfin, cette recherche contribue trois traqueurs d'objets : CTSE, TUNA et FaceTrack, respectivement. Les expériences montrent que le système proposé est assez robuste à de nombreux défis dans le suivi du visage (objet) visuel et a une stratégie bien établie pour effectuer la mise à jour en temps opportun du modèle d'objet en ligne, ce qui le rend plus efficace pour une application dynamique comme le suivi. La méthode proposée surpasse méthode de suivi de la littérature. Le code est écrit en C ++ avec une dépendance sur la bibliothèque open source fournie par OpenCV Community. Le code est ouvert pour faciliter la recherche et le développement.

ABSTRACT

This work presents a detailed analysis and discussion of a novel face tracking system that utilizes multiple appearance models along with a tracking-by-detection framework that can aid a video-based face recognition system by giving face locations of specific individuals (Region Of Interest, ROI) for every frame. A face recognition system can utilize the ROIs provided by the face tracker to get accumulated evidence of a person being present in a video, in order to identify a person of interest that is already enrolled in the face recognition system.

The primary task of a face tracker is to find the location of a face present in an image by utilizing its location information from the previous frame. The searching process is done by finding the best region that maximizes the possibility of a face being present in the frame by comparing the region with a face appearance model. However, during this face search, several external factors inhibit the performance of a face tracker. These external factors are termed as tracking nuisances, and usually appear in the form of illumination variation, background clutter, motion blur, partial occlusion, etc. Thus, the main challenge for a face tracker is to find the best region in spite of frequent appearance changes of the face during the tracking process. Since, it is not possible to control these nuisances. Robust face appearance models are designed and developed such that they do not too much affected by these nuisances and still can track a face successfully during such scenarios.

Although a single face appearance model can be used for tracking a face, it cannot tackle all the tracking nuisances. Hence, the proposed method utilizes multiple face appearance models. By doing this, different appearance models can facilitate tracking in the presence of tracking nuisances. In addition, the proposed method, combines the *tracking-by-detection* methodology by employing a face detector that outputs a bounding box for every frame. Therefore, the face detector aids the face tracker in tackling the tracking nuisances. In addition, a face detector aids in the re-initialization of the tracker during tracking drift. However, the precision of the tracker can further be improved by generating face candidates around the face tracking output and choosing the best among them. Thus, in the proposed method, face tracking is formulated as the face candidate that maximizes the similarity of all the appearance models.

While all of the above aforementioned tracking mechanisms are being carried on during the face tracking process, it is essential to timely update the appearance models to cope up with the face appearance changes. The update acts like a temporal memory that accounts for the appearance changes. Hence, a timely update of model requires a strategy that needs to be cautious, otherwise an untimely update will result in contamination of the model. In our proposed method, we perform two types of updates: long-term appearance update and short-term appearance update. The long-term appearance update keeps reliable appearance features that comes handy for precise face localization. Hence, these updates happen during the entire video sequence. On the other hand, the short-term appearance update helps to track the face during sudden abrupt appearance change due to tracking nuisances. This helps to avoid potential tracking failures such as drift. The short-term updates are governed using an occlusion detection strategy and by analyzing the precision of a feature for face location.

Finally, three contributions are made during the course of this research project: CTSE, TUNA and FaceTrack, respectively. The experiments show that the proposed methods show robustness to numerous challenges in visual face (object) tracking and has a well-laid strategy to perform the crucial online object model update, thus making it more efficient and effective for a dynamic application like tracking. The proposed methods outperform several state-of-the-art object trackers. The code is written in C++ with a dependency on the open source library provided by OpenCV Community. The code is open-sourced to facilitate research and development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ROI	Region of Interest
FAST	Features from Accelerated Segment Test
CenSurE	Center Surround Extremas
FREAK	Fast Retina Keypoint
LBP	Local Binary Pattern
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
BRISK	Binary Robust Invariant Scalable Keypoints
BRIEF	Binary Robust Independent Elementary Features
AM	Appearance Model
BB	Bounding Box
FR	Face Recognition
CNN	Convolutional Neural Network

CHAPTER 1 INTRODUCTION

The ability of a system to robustly locate objects of interest in images and videos has numerous enticing applications that the modern technology is thriving for. For example, finding a face automatically, tracking it and then augmenting it with a filter that adds elements such as hat or goggles, has changed the experience of people sharing photos. Self-driving cars can automatically detect and track other cars and objects which the human drivers sometimes fail to locate themselves. Face recognition systems can automatically detect and track individuals entry/exit to a particular building or a place of interest. Places like airport, security checkpoints, and scientific laboratories are already benefiting from the current technology of recognition and surveillance systems. Further, human interaction with a virtual environment and augmenting data over the top of the real-world have only become possible through the recent developments in the field of computer vision tracking, detection, graphics, machine learning, and computer hardware that can sustain the complexity of such systems.

1.1 Motivation

With the plethora of existing applications and more upcoming in the future, face tracking and face recognition systems are becoming popular due to their increasing demand in surveillance and security systems. Their premier objective is to first detect faces and then identify them in the video based on the persons that are already enrolled in the security system. To achieve this objective, the video frames go through a series of face processing tasks such as face detection, face matching and face classification that may utilize the *spatio-temporal* information from the frames. In order to utilize this *spatio-temporal* information, several FR systems like [12], [13], and [14], employ face trackers to output facial regions of interest (ROIs) of a person present in the video (Refer Figure 1.1). This group of ROIs that correspond to a face *location* over a certain period of *time* is called the spatio-temporal information.

1.2 The role of tracking in video-based face recognition

Consider the scenario of security checkpoints, where security agents that are currently manually identifying people by direct visual contact, would be assisted by a video-based FR system. In such a system, where the main task is to recognize a face in an *online* (not necessarily *real-time*) manner is very challenging. This is because, unlike a controlled capture environment of a *mugshot* (upfront face, face clearly visible) being taken, the capture condi-

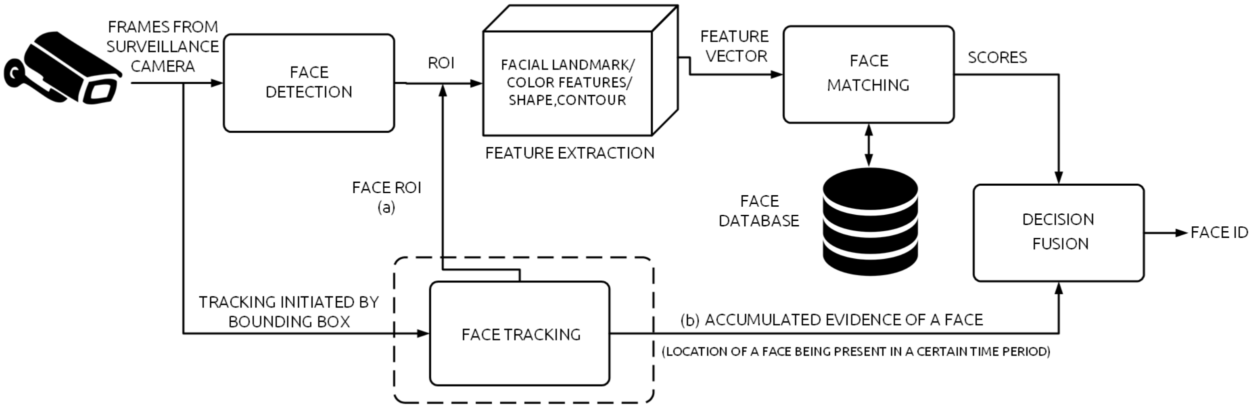

FRAMES FROM SURVEILLANCE CAMERA
FACE DETECTION
ROI
FACIAL LANDMARK/ COLOR FEATURES/ SHAPE,CONTOUR
FEATURE EXTRACTION
FEATURE VECTOR
FACE MATCHING
SCORES
FACE DATABASE
DECISION FUSION
FACE ID
FACE ROI (a)
TRACKING INITIATED BY BOUNDING BOX
FACE TRACKING
(b) ACCUMULATED EVIDENCE OF A FACE
(LOCATION OF A FACE BEING PRESENT IN A CERTAIN TIME PERIOD)

Figure 1.1 A generic video-based Face Recognition System. A Face Tracking system (shown by dotted lines) provides ROI and multiple accumulated evidence of a person of interest over time being present in the video to assist a Face Recognition System.

tions are dynamic and cannot be relied on a certain set of poses and expression. In addition, there are several constraints like illumination change, low resolution of the video, fast motion of the face, face deformation due to expression change, pose variation and scale change of the face when moving away or towards the camera, partial occlusion and full occlusion due to the presence of other non-targets, false positives due to the presence of non-target faces in the environment, etc. Dealing with such kind of challenges requires spatio-temporal information which can be obtained only by using a Face Tracking system. Robust face tracking algorithms have been used to enhance the capability of the FR system by providing ROI over a number of video frames that can be used further by the FR system to extract relevant features for face recognition. Hence, a face tracking algorithm provides a coarse estimation of where a face is located in the video frame and assists the video based FR system. Once a face is located, the ROI must be extracted for features and converted into a form required for face recognition. Thus, the role of a face tracking is to provide assistance to a video based FR system by providing :

- ROIs for the faces at the end of each frame processed by the face tracking system (refer Figure 1.1 arrow (a)), from which relevant features for video-based FR can be extracted.
- multiple accumulated evidences of spatio-temporal information of a face present over a length of video sequence provided by the face tracking system. The tracking algorithm accumulates spatial and motion (temporal) information of the faces present during the video sequence by continuously finding image regions based on the prior observations of the tracked faces in the video sequence (refer Figure 1.1 arrow (b)). Thus, this

accumulated evidence of the observation of a same face can assist the video-based FR system to make more confident and reliable identification.

This spatio-temporal information provided by the tracker ensures the conformity of each face's identity due to feedback provided by the tracking system. That is, the identity of the person can be validated over many frames. Hence, this work focuses on the face tracking in real-world unconstrained video sequences. Nevertheless, there have been tremendous efforts to solve the problem of face tracking, but a lot more is still left to solve it completely. This thesis aims to propose solutions to solve the face tracking challenges in a robust, stable, adaptive, self-driven (semi-supervised learning) and intuitive manner that can assist a face recognition system by providing regions of interest over the duration of a surveillance task being carried out.

1.3 Problem Statement and Scope

One of the major challenges while tracking a face in a video sequence is the uncertainty of its position in a frame. The uncertainty arises because the face undergoes various appearance changes due to its interaction with the surrounding environment like change in illumination, pose, occlusions, deformation, etc. The following subsections discuss the face tracking problems in detail.

1.3.1 Dynamic appearance change of face, unreliable face detection and low-resolution video quality and computational efficiency

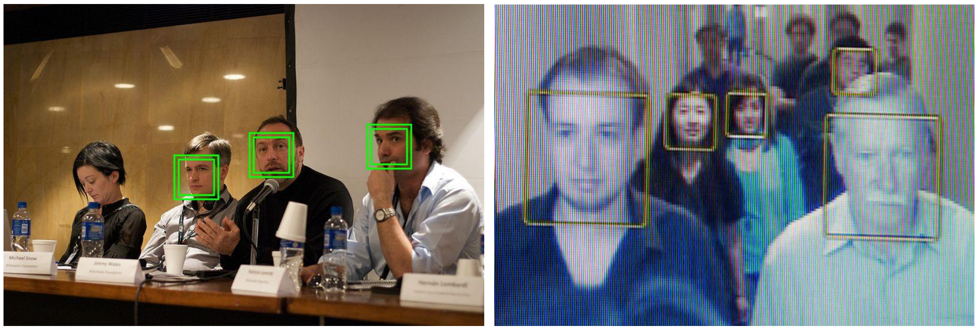
In real-world unconstrained environment, tracking a face in a video is very difficult due to two major difficulties. The first difficulty is the ability to detect the faces in an unconstrained environment, so as to discriminate the faces from the background and also to distinguish them from other objects having similar appearance, i.e. from other faces. And second is the unreliable detection from a face detector that fails with frequent changes in the appearance of the faces due to dynamic environment conditions like change in illumination, which in turn causes change in the color and texture of the face. Face detection also fails because of the visibility issues such as a view of the full frontal face does not appear to the camera as shown in Fig. 1.2a or when the faces become occluded when they move in a crowd. This frequent changing of the appearance model causes miss and false detections and hence unreliable tracking [15]. Face detection failure can also be caused by the changing scale of a face while entering and exiting the surveillance environment (See Figure. 1.2b). The faces may appear abruptly at any time, anywhere and at any scale in a scene. In this scenario, the dimension of a face does not remain constant for every frame, due to which the detection results are unreliable. Thus, it is very hard to train a face detector to work at all scales. This problem further extends when the face undergoes deformation due to expression and pose change. In such a situation, confirmation of a face becomes very difficult and hence their maintenance of the unique identity for a certain duration of video sequence. Usually, the surveillance videos are of low-resolution. Hence, the features chosen for tracking should be machine detectable for such type of videos and not only constrained to videos that are of high quality. Since, the tracking application is dynamic, it demands rapid processing of frames and output generation. Hence, the computational efficiency of tracking should be kept in mind and should be suitably developed towards online performance.

1.3.2 Keeping track of the inter-frame correspondence of the target face among distractors

Another difficulty is establishing the inter-frame correspondence of a face of interest for every frame. During this, the most probable tracking result has to be assigned to the target face, and not to the other faces that are present in the surrounding. As there are objects having similar appearance to the target face and its appearance properties change due to the real-world unconstrained situations, there may occur target identity switching of the tracking output between multiple faces. Furthermore, this becomes quite frequent with the increase in the number of faces in the scene and becomes very difficult to correct during the tracking process. Thus, the inter-frame correspondence may not remain distinct to each of the target face and the tracker might drift away from the target, which may result in tracking failure [15].

1.3.3 Tracker Initialization

It is a process in which a region of interest (usually a bounding box) is marked for the tracker that gives information about the location (position and size) of a target of interest at the beginning of a video sequence. This information is used as an initial estimate for the tracking process for the rest of the video sequence. In tracking, the problem of localization is different from object localization in a static scene. In tracking along with matching the appearance similarity of the object between two consecutive frames, a correlation of the matched features between two consecutive frames is also required. The correct tracking result is inferred from information (temporal/motion) in previous frames, since the position of the object in the next frame is normally constrained to a specific region. Whereas, in a static scene, the object of interest can be present anywhere and is localized (detected) by only appearance similarity



(a) (b)

Figure 1.2 (a) Missed detection caused when the full frontal face view is not visible to the camera.(This image is licensed under the Creative Commons Attribution 3.0 Unported License.) (b) Failed detection due to different scale of faces. (This image is adapted [reprinted] from [4] © 2011 IEEE).

of the object. It does not matter where the object was present before in the previous frame. In static scenes, the appearance is learned offline from many examples, while in tracking learning is *one-shot* and starts with a single example. Thus, in tracking both the appearance and motion information is used to localize the target. Hence, correct initialization of the tracker by a bounding box is crucial for best tracking result in the next consecutive frame. Usually, this is done by a manual or an automatic method. In a manual method, the initialization is done by the user by annotating the object in the first frame. In an automatic method, the initialization is done by using object detection algorithms, for example, by using face detector [10], [16] or a human detector [17]. Majority of tracking algorithms are initialized manually and assume that the object information is accurate. But, in case of bad initialization which can be manual (due to human error) or automatic (output by a object detector) as shown in Figure 1.3a, the tracking results may differ for the same video sequence, which leads to performance loss (refer 1.3b). Thus, the tracking process can be highly sensitive to initialization and is a very difficult problem to solve.

1.3.4 Appearance Model Update and dealing with drift

In order to compensate for the changes in the appearance model due to dynamic environment conditions, updating the appearance model becomes necessary from time to time. During this model update process, the tracker sometimes drift away from the target. The drift is the phenomenon where the tracker gradually shifts away from the target object during the tracking process and eventually results in loss of target object. The main reason behind drift is the accumulation of errors, when the appearance model is updated in the *absence* of

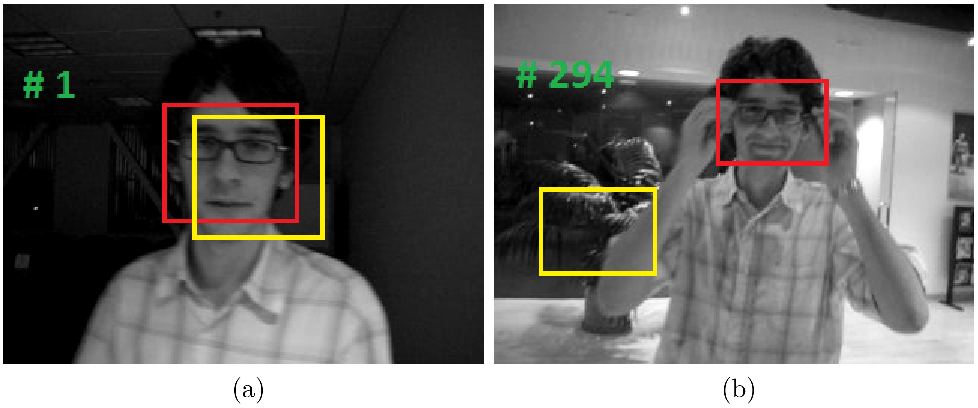


Figure 1.3 Tracking result due to bad initialization leading to performance loss (Red Box : Ground Truth, Yellow Box : Bad initialization). (a) Initialization by Yellow Box at first frame. (b) Tracker result represented by yellow box at 294th frame. Since the captured appearance from the yellow box in the first frame includes a lot of background, the tracker is distracted from the real target. (These images are adapted [reprinted] from [5] © 2011 IEEE.)

ground truth. By updating, the tracker improves its prediction for the future frames but not necessarily all the time, as there is always a possibility of contamination of the appearance model of the target. But, in real world unconstrained videos, the tracker over time adapts itself to distractors (objects with similar features to the target object). In tracking faces, the tracker may adapt to other similar faces in its surrounding. Results in literature have shown that very frequent update of appearance model results in contamination of the appearance model and thus large drift [18]. On the other hand, if the appearance model is not updated at all then tracking failure also occurs as shown in Figure 1.4. Therefore, in order to avoid drift, the rate of appearance model update has to be carefully chosen depending upon the quantitative analysis of the tracking quality so that a good trade-off can be achieved.

1.3.5 Abrupt motion and long-term occlusion

Sometimes the motion of the target object is very fast or the target appears abruptly during the video sequences, which causes uncertainty in tracking results. The tracker searches in the neighborhood region of the target object based on its location in the current frame and predicts the output in the next frame according to this search. During this abrupt motion change, the target object might appear outside this search region, due to which the localization of the target object in the next frame is incorrect and hence results in tracking failure (See Figure 1.5). In scenarios where the target object disappears and reappears in the camera view or is occluded by another object for a certain time and reappears after sometime at a different location, the tracking problem becomes more complex. Here, the

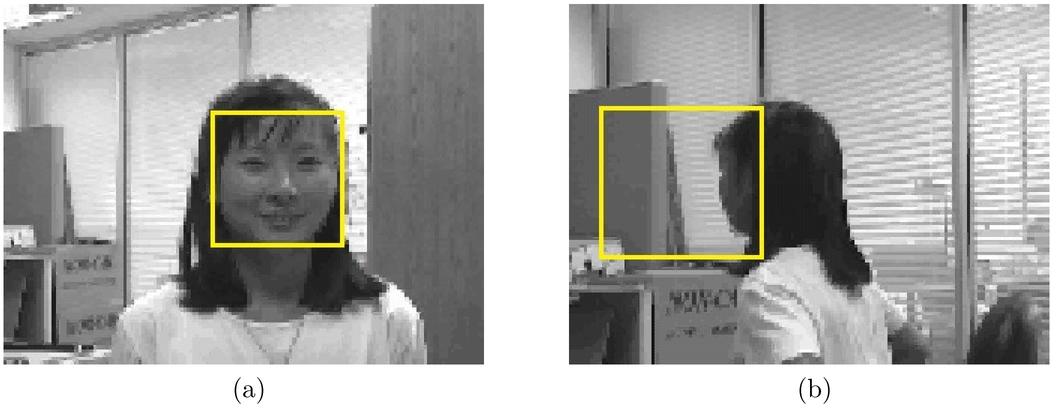


Figure 1.4 Tracking Failure when the appearance model is not updated. (a) Tracker Initialization in Frame 1. (b) Tracking failure in later part of video sequence when the appearance model is not updated. Here the object's appearance changes due to pose variance. (These images are adapted [reprinted] from [6] © 2011 IEEE.)

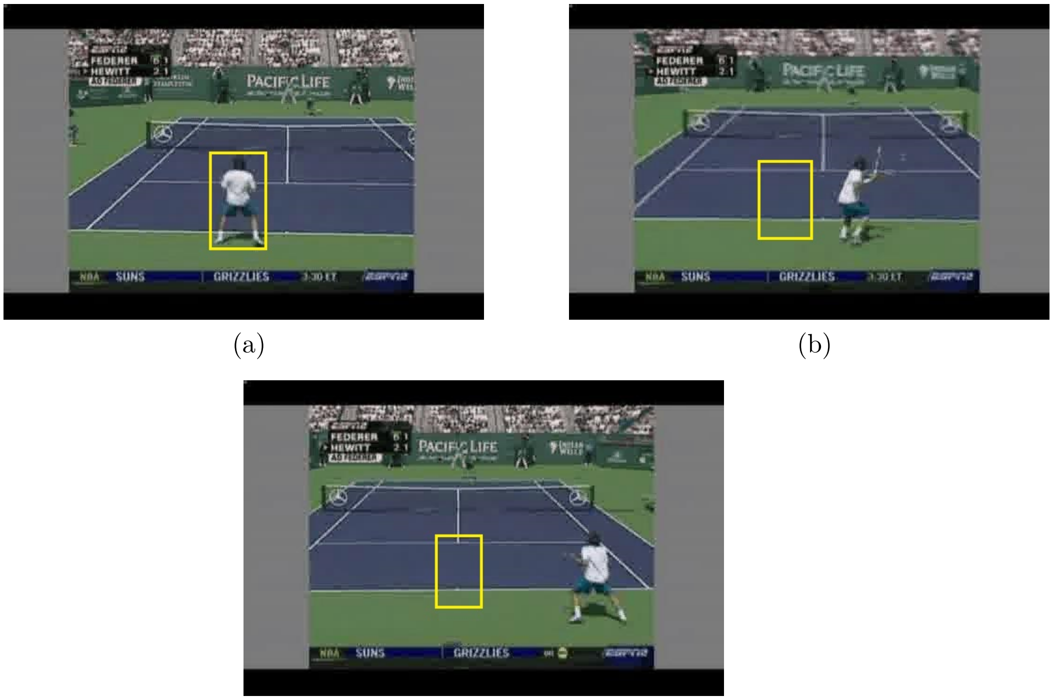
tracker should be able to associate the same output label to the target object before and after the interaction, but it becomes more tedious with the increase in number of target objects along with frequent appearance changes.

Thus, in the process of face tracking in videos, all of the above mentioned problems are caused due to the frequent appearance change of the face in real-world unconstrained settings, which involves both photometric (color, texture) as well as geometric (face deformation, pose, abrupt motion) appearance changes. Hence, for robust tracking, the face appearance model should enable its discrimination from background and also from other similar objects. For this reason, object appearance description is the most important cue for a tracker. But as we have seen, the appearance of the object changes drastically in a real-world surveillance environment, and hence, it becomes imperative to get multiple cues from the appearance of the object.

1.4 Research Objectives

The main objective of this thesis is to develop a strong appearance model for tracking of faces in unconstrained video sequences that tackles specifically distractors, low-resolution, face deformation, and fast motion, during tracking. The tracking system should provide accurate face location in the image with least localization error. Thus, the specific objectives of this research are identified as the following :

— To develop a robust appearance model by identifying features that are stable in real-world unconstrained video scenarios particularly dealing with low resolution videos,



(a)

(b)

(c)

Figure 1.5 Abrupt motion change results in tracking failure. Notice that the position of the tennis player changes very fast between the consecutive frames (a) - (c). (These images are adapted [reprinted] from [7] © 2011 IEEE.)

video sequences having large variations like motion blur, occlusion, illumination without compromising the performance of a tracking application.

— To develop a tracking strategy which is robust to the initialization of the tracker. It should also incorporate the inherent tracking noise and deformation while tracking a target face.

— To develop a control strategy that provides stable face tracking in spite of distractions present in the scenario. i.e., to develop a strategy to prevent tracking drift by preventing the tracker to not to get influenced by the false positives produced by a detector i.e., to discard the region identified as a potential face location even when a face is not present.

— To develop an update strategy to perform reliable appearance model update during online tracking in the *absence* of ground truth.

The following paragraph discusses the main contributions to each research objective.

1.5 Contributions

In this work, our objective is to develop a robust and accurate face tracking system for tracking faces in a video surveillance environment. This is done by considering the face tracking problem as a generic object tracking problem. For this purpose, our *first* contribution is the development of a model-free tracker, which is presented in Chapter 4 (paper published in ICIP, 2015). The tracker is robust towards the fast motion of the object, partial occlusion and deals with distractors present in the surrounding. The tracking features are evaluated for their quality online during the tracking process and the latent features are discarded from the appearance model. A novel strategy is proposed to deal with the inherent tracking noise during the process of tracking in video sequences.

In Chapter 5, the *second* contribution consists in developing a model-free tracker to tackle scale change of the target object (paper submitted to MVA, July 2016). Further, multiple cues of the target object are utilized as features, which makes the tracker more robust towards distractors, low-resolution, and helps in tracking the target object over long periods in video sequences.

Finally, the *third* contribution is presented in Chapter 6, by making the tracker specialized for tracking faces by proposing multiple appearance models and incorporating a face detector with the face tracking system (paper submitted to Image and Vision Computing, June 2017). This is done because neither detection nor tracking *alone* can solve the complex challenges that occur while tracking faces. The advantage of using a face detector aids in adding prior information about the target face and can handle abrupt motion changes and scale of the target face, but cannot handle occlusions very well. On the other hand, the advantage of using a face tracker is that it addresses the problem of face appearance matching which becomes challenging in real-world unconstrained situations. The multiple appearance models can handle partial occlusions, and several other tracking challenges. For improved localization of the face tracking output, the face candidates are generated around the localized face. Finally, a weighted score-level fusion approach is proposed for selecting the best final output. The operation of the above mentioned components are independent and complement each other for robust face tracking.

1.6 Thesis Structure

The structure of this thesis is as follows : Chapter 2 gives a brief summary of the state-of-the-art methods in visual object tracking. Chapter 3 gives the overview of the proposed approaches in direction of solving the challenges in the research project. The next three

Chapters discusses about the progression and contributions of the research, and is presented by introducing the articles in Chapter 4, Chapter 5 and Chapter 6 respectively. Chapter 7 provides a general discussion of the proposed solutions and the impact of research in the domain of surveillance, computer vision, and machine learning. Finally, thesis is concluded with Chapter 8.

CHAPTER 2 LITERATURE REVIEW

In this chapter, we will define and discuss about the tracking process and the related terms used in object tracking. This chapter also focuses on analyzing the advantages and in identifying the shortcomings of the popular state-of-the-art techniques in visual object tracking literature.

2.1 Object Tracking

Object tracking can be defined as the task of locating an object of interest in a given video sequence. The input to the tracking system is a frame and the output is the location depending on the type of 2-D (x,y) or 3-D (x,y,z) tracking system. The 2-D output is usually represented by a rectangular bounding box encapsulating the object for a given frame. Some other popular shapes are ovals, circles, etc. For 3-D, object pose estimation may also be provided along with location of the object.

Quoting from a survey paper [19], ”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”. Different interpretations of tracking an object are derived in the literature. Globally, tracking can be considered as the following tasks :

- It can be considered as an estimation of state for a certain period of time for a state space model.
- It can be considered finding the minimum distance between the tracking state and the subspace appearance model which is learnt during training or where the training data is limited (For example, from a single frame, or from a number of patches). Usually generative appearance models follow this tracking strategy. Here, they do not incorporate any background information present in an image.
- It can be considered as a classification problem where a classifier is trained to distinguish the target from the background and distractors. Usually, discriminative appearance models follow this concept of tracking the target object.

In order to track an object effectively, strong features are required by the tracker. Typically the features, which describe *unique* and *distinct* characteristics of the object, are particularly chosen, because the tracker may confuse the same target object as some other object and hence may loose track or drift away from the target object. Thus, features from the target object that do not change rapidly and are robust to the uncontrolled environment changes are crucial for tracking. For successful tracking operation, the features having the following desirable characteristics are usually preferred :

- The features are *discriminative* and describe the unique properties of the target object.
- The features that occur frequently, are *repeatable* so that they can *facilitate* tracking.
- The features that are *spatially specific* as the primary goal of tracking is to get precise location about the target object.
- The features that are detected by the machine easily over sophisticated ones that might take more detection time during *feature extraction*.

Therefore, strong features that are kept as reference for feature matching is called an *appearance* model of the tracker. The next section describes the process of tracking an object in a video sequence and the important components of a tracking system.

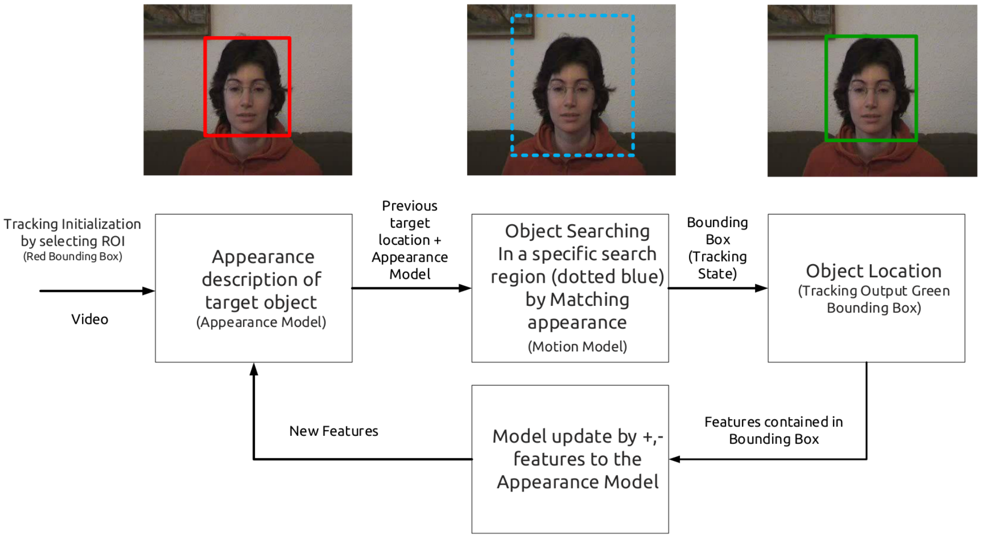

Tracking Initialization
by selecting ROI
(Red Bounding Box)
Video
Appearance description of target object
(Appearance Model)
Previous target location + Appearance Model
Object Searching In a specific search region (dotted blue) by Matching appearance
(Motion Model)
Bounding Box (Tracking State)
Object Location
(Tracking Output Green Bounding Box)
Features contained in Bounding Box
Model update by +,- features to the Appearance Model
New Features

Figure 2.1 Object Tracking Process

2.2 Visual Object Tracking Process

A tracking algorithm includes two main components : (1)an appearance model that represents the characteristics of the target object, and (2) a search strategy to estimate the target's position in every frame. As seen in Figure 2.1, the tracking process is first initialized by annotating a bounding box around the ROI that is required to be tracked. Depending on the type of appearance model, the important features of the ROI comprise the appearance model. After this, for the next frame, the tracking process consists of first detecting features in a frame and the *feature matching* is done by comparing the detected features with those present in the appearance model for similarity. Typically, in most of the tracking algorithms, the features are detected and matched in a specific search region around the target's previous location as shown by blue dotted lines in Figure 2.1. This is because the target's motion is usually limited between two consecutive frames. The features that have the highest similarity after feature matching are considered as possible indicator of the target locations by taking their spatial information (x and y coordinate in image plane). Various strategies are implemented by tracking algorithms to arrive at the best possible target location. As an output, a bounding box is given as the final target location.

Since the target appearance may change abruptly during tracking, as discussed previously in Section 1.3, the features contained in the appearance model might not match for similarity during such scenarios. Hence, timely addition, modification and deletion of features are performed during the appearance model *update*. All of the above operations happen online (real-time) while the process of tracking an object continues during the video sequence. Updating the appearance model is crucial for tracking (during the *absence* of ground-truth), as untimely update can result in tracking drift or track loss of the target object. The following section provides an overview of the appearance representations that are used in object tracking.

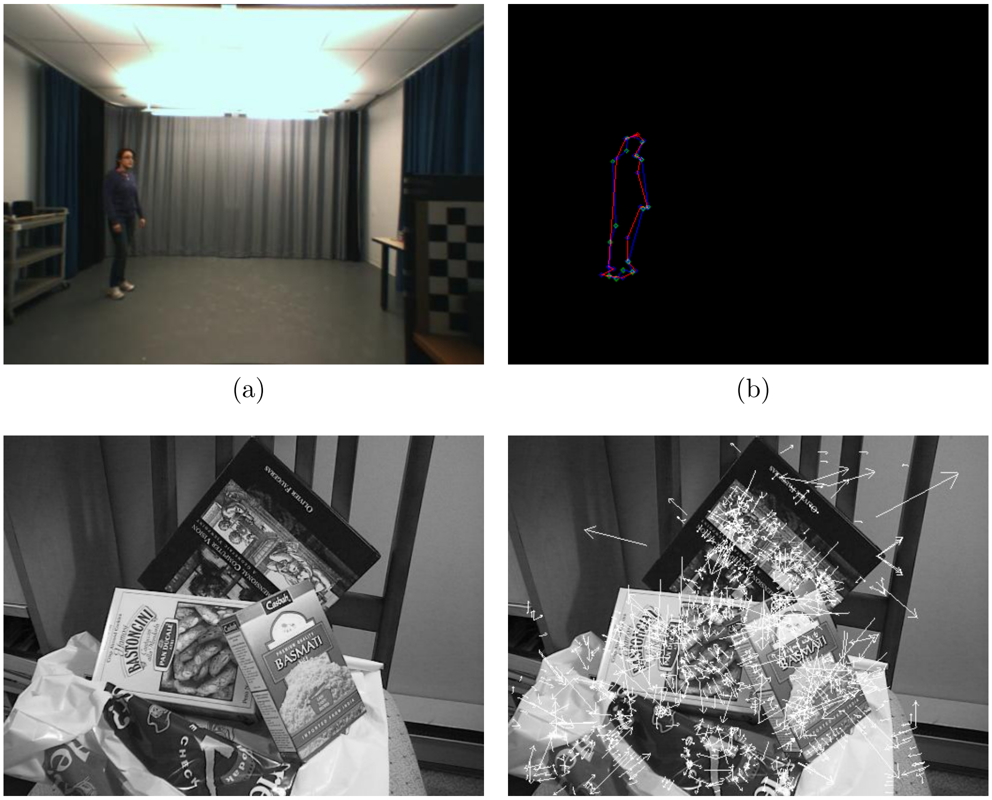
2.3 Appearance Model in Object Tracking

The target object appearance description is the most important cue for tracking in video sequences. The appearance model is the target object representation, which comprise of characteristic features of the object. It is a way of injecting prior and useful information about the target object (object of interest) for tracking. It enables its discrimination from distractors (similar looking objects in the surrounding) and assists in achieving frame to frame correspondence during the tracking process. It is evident from the literature that there has been a tremendous research effort on modeling the appearance of a target object and numerous representations have been proposed in [19], [20] and [21]. If we take a closer look at all of these object representations, they can be broadly classified as *generative* and *discriminative* appearance models. In generative approach, only the appearance of the target object is taken into account for modeling, and the main focus is to find the region that is the most similar to the target object's appearance model. This approach does *not* require a large amount of training data. However, it utilizes the information obtained from both the *previous and current frame*, for estimating the target object location in the next consecutive frame. In discriminative approach, the target object is modeled in *relation to the surrounding environment*. Here, tracking is considered as binary classification task that tries to separate target object's appearance and background regions. This approach utilizes only the appearance information in the *current* frame for estimating the target's location in the next consecutive frame. However, this approach also requires a large number of examples for training the binary classifier, so as to distinguish effectively the target object and others. The following subsections discuss in detail about the advantages and shortcomings of the feature representations used in object tracking.

2.3.1 Interest Point Model with Feature Descriptor

In this method, the target object is represented using interest points, as shown in Figure 2.2b. An interest point (or keypoint) is a point, which is invariant to illumination, viewpoint, scale and rotation. It is a local extrema (having maximum or minimum value), which is found at different scales of an image, re-scaled many times, when different widths of Gaussian smoothing kernels (or any other kernels) are applied to the image [22]. Thus, these points are very interesting for tracking, as they are repeatable and can be found on a target object, even when its size changes abruptly during the video sequence. The interest points are first detected by point detectors such as Harris [23], CenSurE [24], FAST [25], SIFT [22], etc., and then local neighborhood information around the interest points is encoded. Here, the point correspondence is achieved by using a distance measure, such as the $L1$ norm or $L2$ norm (*Euclidean distance*), which is a similarity measure based on proximity. Sometimes, the pixels are grouped into a vector and matching is performed by comparing the vectors using *Bhattacharya* distance.

The interest point model is obtained in three steps : (1) Interest Point Detection, (2) Feature Description and (3) Matching Feature Descriptors. The interest points in a frame are first detected by using point or edge detectors. Usually corners of an object are chosen as interest points, since they are stable and invariant to illumination, rotation and viewpoint. But, if we only use the spatial information about the corners for matching, it is possible that the object shifts abruptly in the next frame, and thus, matching the same points may not be possible due to environment factors, change of viewpoint, occlusion etc. Therefore, to avoid this ambiguity, local neighborhood information is also embedded for matching, by analyzing textural information around the interest points and describing them using feature descriptors like SIFT [22], SURF [26], FREAK [27] etc.. Feature descriptor is a high dimensional vector containing a set of local neighborhood measures around the interest point. Here, the frame correspondence is found by comparing the similarity of feature descriptors between the candidate region of interest and that of target object appearance model using a distance similarity measure like *Euclidean* distance. The feature descriptors are computed separately for the candidate region and the target appearance model, respectively (See Figure 2.2).



(a)

(b)

(c)

(d)

Figure 2.2 Interest Point Appearance Model. Target object represented by interest points. (a) Original Test Image (b) Interest points representation on the contour [8]. (c) Original Test Image (d) SIFT feature descriptor output for test image (c). Here the length of arrow indicates the proportional scale of an interest point detection, and the direction indicates the dominant orientation of an interest point. (These images are adapted [reprinted] from Demo Software : SIFT © 2005 University of British Columbia)

The local neighborhood information is used for computing the feature descriptor. The feature descriptors use pyramid techniques (*Scale Space*) to detect the most characteristic interest point at many scales in a frame, which therefore aids in locating the target object appearing at all scales in a frame. Further, the interest points are localized (distinct and non-ambiguous) and are robust to illumination, scale, occlusion and viewpoint. Since, these points occur in abundance, due to large textural information in an image, their repeatability rate is often large, which is desired for better frame matching. The model is also robust to occlusion, as there are enough interest points available for matching, even if a larger part of the target object gets occluded in the scene. Hence, the advantage of using interest point model is that it can be used to represent very small objects. It is very simple and fast to compute and hence can be used in applications where fast tracking is required. Further, binary feature descriptors like LBP [28], BRIEF [29] and BRISK [30] are also popular features. They give similar performance to the non-binary feature descriptors like SIFT but are even faster in computation and require less memory as compared to them. However, the interest point models do not provide geometric information about the target object. Since they exclusively describe the target object region using descriptors, the model by its nature comes under the category of a generative appearance model.

2.3.2 Region-based Model

In this method, the appearance of the target object is modeled using all the information of pixels contained in a target region, which is usually depicted by a bounded box, silhouette, patches, etc. The information such as texture, color and template-based sparse description are used as matching criteria for frame correspondence. Seldom, instead of modeling all the pixels inside the bounding box, some authors try to model only the pixels belonging to the target. In that case, for extracting the target region in a frame, approaches like segmentation and geometric outline of the target object are used as matching criteria for frame correspondence. Given below, is a list of target region model that is used in object tracking.

1. Geometric Shapes : The object is modeled by using geometric shapes such as rectangle, square, circle and ellipse. For example in [6], the head is represented by an ellipse model for tracking the head of a person. Generally geometric shape models are used for tracking rigid objects, which do not undergo extreme geometric changes. Similarity measures such as proximity, length and symmetry are used for frame correspondence.
2. Color : It captures the visual information contained inside by the object region. Histogram is used to model the color cue of the object. It represents the number of occurrences of each color that is present in the object region. The histogram can be single channel as in the case of gray scale images or multi channel such as RGB (Red, Green, Blue) or HSV (Hue, Saturation, Value). Here, the color histogram is calculated for each color channel and is later combined to form a three-channel histogram. The frame correspondence is achieved by matching the candidate region color histogram with the target object color histogram using *Bhattacharya* distance as is done in mean-shift tracking [31]. The advantage of using histograms is that they are invariant to small illumination changes and relatively invariant to translations, rotations, partial occlusion and scale changes. The disadvantage of histogram description is that they cannot efficiently capture the spatial arrangement of the intensities, i.e., they are

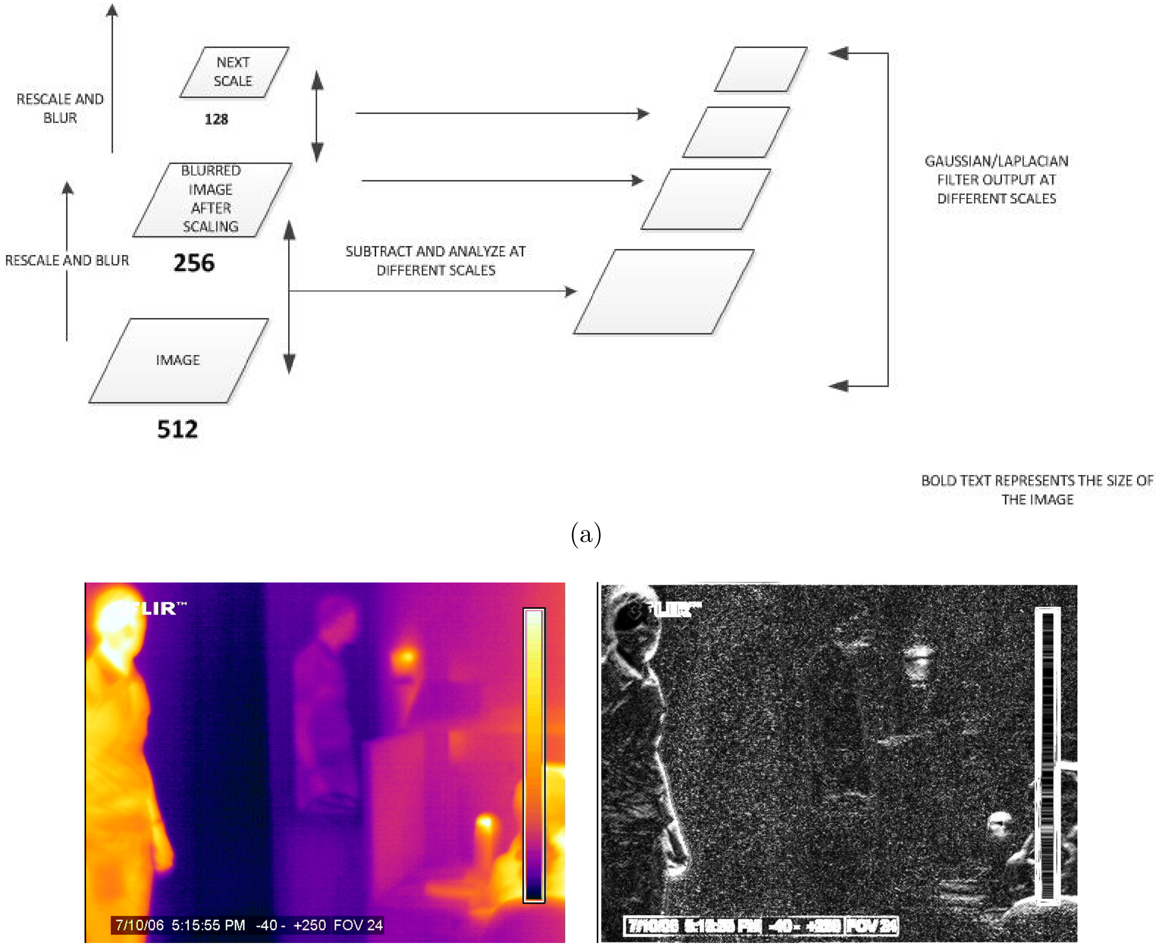
unable to capture textural information present in a region, due to which they may produce ambiguous matching and thus tracking results may suffer.

3. Texture : It gives information about the patterns present on the surface of the object. The patterns represent the spatial arrangement of intensities in an image. It takes into account the relative position of the intensity gradient present in an object region. To extract the pattern information, the image is convolved in x and y directions using gradient filters, such as Sobel or Prewitt filters [32]. These specialized filters assist in detecting gradient orientation on the object's surface in both x and y directions. The various gradient orientations are then concatenated into a histogram. The strength of each gradient orientation in the histogram corresponds to the number of times a particular edge orientation occurs in an image. To describe a quantitative information about the density of edges in a texture, the magnitude of all the gradients contained in a region, are also considered along with their orientation [33].

   Another technique to describe the quantitative information about the edge in an image region, is by convolution with filters that resemble patterns, e.g. bars and spot filter. These filters are weighted combination of symmetric gaussian filters with different sigmas to detect bars (particular gradient orientation) and spots at different scales and orientations in an image.

   A very popular technique to analyze different textures occurring in an image is by using pyramid techniques which is also known as *Scale Space* technique [32]. The popularity of pyramid techniques is due to the fact that they can be used to detect orientations at different scales in an image. As the target object can appear at different scale (size) in an image and therefore to detect it, it is important to know the dominant orientation that is prevalent in a particular frame. For constructing a pyramid, first the image at a particular scale is convolved with a Gaussian kernel and then the same image is down-scaled in both x and y direction and again convoluted with a Gaussian kernel to analyze its corresponding response. However, this results in redundant information. Thus, to remove this redundant information occurring at different scales, difference of Gaussian is calculated between the two versions of the Gaussian kernel's response (Refer Figure 2.3). Hence, the advantage of using textures for an appearance model is that they are often less sensitive to illumination change and more discriminative than simple color models.

4. Sparse Representation : In this, the target region is represented as a linear combination of a set of templates [9]. The templates are of two types : target and trivial templates. The trivial templates are used for handling occlusion and are sparse in nature, i.e. these templates only have one non-zero element (See Figure 2.4). The target templates


RESCALE AND BLUR
NEXT SCALE
128
BLURRED IMAGE AFTER SCALING
RESCALE AND BLUR
256
IMAGE
512
SUBTRACT AND ANALYZE AT DIFFERENT SCALES
GAUSSIAN/LAPLACIAN FILTER OUTPUT AT DIFFERENT SCALES
BOLD TEXT REPRESENTS THE SIZE OF THE IMAGE
FLIR™
7/10/06 5:15:55 PM -40 - +250 FOV 24

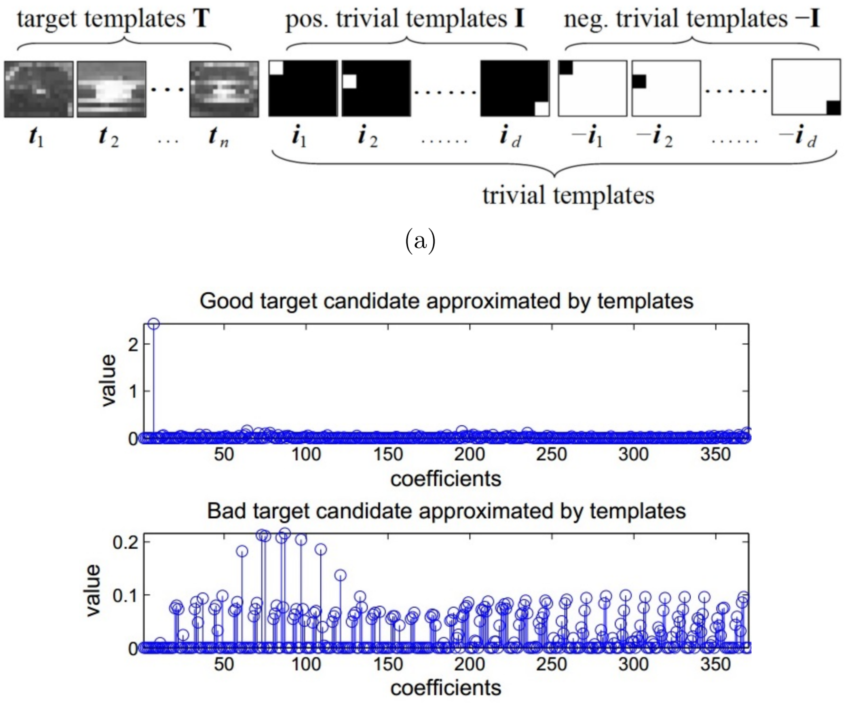
(a)

(b)

(c)

Figure 2.3 DoG (Difference of Gaussian). (a) *Scale Space* DoG Pyramid Construction (b) Output Response of the filter. See the gradients that are clearly detected in the image.

are the templates associated with observations of the target. A good target region is one that requires the least of the target templates from the template set for its reconstruction. If the candidate region is a combination of the trivial templates from the template set, it is considered as a bad region. The advantage of this model is that it is more robust than a template matcher, since it considers the template with the least reconstruction error as a good region. The model is able to handle gradual scale changes of target object's appearance and can also deal with changes in viewpoint. The disadvantage of this model is that the template representation is sensitive to illumination change, not very robust to occlusion and in tracking deformable objects, because the whole image region is represented as a template, unless the templates are represented as patch regions. By its nature, the sparse model is a generative appearance model as it gives description about the whole template.



(a)

(b)

Figure 2.4 Sparse Representation (a) Target Region represented by a set of combination of target and trivial templates. (b) A good candidate is the one that will have least number of trivial templates, i.e the set of templates will have zero coefficients will lead to a sparser representation (Notice the coefficient values in the graph. The good target candidate will have very low coefficient values unlike the bad target candidate. (These images are adapted [reprinted] from [9] © 2011 IEEE)

5. Patch Model : In this model, the appearance of the target is represented by patches (sub-regions) instead of representing the target object as a whole single entity. This

model is able to retain information of the target's appearance in cases of occlusion, and pose change. Even if a larger part of the target remains invisible or occluded, the target object can still be detected and tracked correctly as some patches will still be visible and can contribute to an important visual cue for tracking. In [34], the appearance model is comprised of multiple patches, which describe the local information of the appearance of the target object. The goodness of each patch is determined by a robustness measure, and based on this measure the patches are updated, i.e., less significant patches are deleted and more significant patches are added online according to the current predicted location of the target object in a frame. This model exhibits some similarity with point-based model with feature descriptor.

2.3.3 Contour Model

In this method, the appearance of the target is modeled by the shape of the target object or by the color or texture inside the outline of the object. Modeling by shape is different from modeling by a target region as there is no background information, since the shape outline is precisely located on the target object boundary.

1. Active Contour Model : This model attempts to minimize energy associated to the target object contour, as a sum of internal and external energy [35]. The contour is represented by a set of points and the associated energy with every point is calculated. This model is also known as a *snake* model.

   The advantage of this model is that it can easily localize to a contour of a target object, so as to achieve a minimum energy state. The disadvantage of this model is that the model requires manual annotation of the target object in the initial frame and assumes overlap between consecutive object positions besides demanding a lot of computations. This model also is a type of generative appearance model.

2. Segmentation Model : Segmentation is the process of dividing image regions having similar properties like color and texture. For example, edge detection can be considered as a form of segmentation, where the target object can be separated from the background based on its boundary(contour). In [36], the pixels which preserve most of the structure of the target object are grouped together as *superpixels* by using graph based Normalized Cuts algorithm [37]. Here each pixel is considered as a node in the graph and the edge between every pair of pixel is weighted by the affinity between the two nodes. The notion behind this algorithm is that dissimilar pixels should be in different segments and similar pixels should lie on the same segment in the graph. The segmented region obtained in the current frame is used as a starting point for

segmentation in the next frame. The advantage of this method is that undefined shaped target object boundaries can be obtained and separated from background without any interference from the background pixels. The disadvantage of this method is that though the method is suitable for single object, it fails to prove robust in case of occlusion and clutter in the background due to overlapping regions of target object position.

All the aforementioned target region appearance models can be categorized as generative appearance models, since they exclusively describe the information contained in the region occupied by the target object. The only exception is the patch model, where the patches can be adjusted to lie on the target object region and also on the background. Hence, the patch model can be used both as generative and discriminative appearance model.

2.3.4 Learning based Appearance Models

These models learn the features of a target object automatically from a set of samples (features) using a classifier and are called as discriminative appearance models. Here, selection of relevant samples (features) directly affects the performance of a classifier. Once the relevant features are selected, different appearances of the target object can be learnt using a supervised learning approach like support vector machines [38]. In [39], patches are generated from the initial ROI in the current frame and is considered as an initial appearance model for tracking. Then these generated samples are classified and labeled as target or background regions using a classifier. The best samples are selected on-line from the current frame and are merged with the initial appearance model, which is used as an appearance model for the next frame. Hence, we can say that these models try to detect the target object in every frame by classifying image regions using the the appearance models. Training discriminative appearance models is a challenging task since they require many parameters. And these must be tuned to get good performance, because if the parameters are not tuned optimally, the tracking might suffer from drift and eventually loss of target [5]. Furthermore, for online adaptation, training samples with correct labels must be chosen effectively for correct target tracking.

Table 2.1 Appearance Models used in Object Tracking

	Challenges in Tracking			
	Scale	**Rotation [1]**	**Illumination**	**Occlusion**
Appearance Models				
Interest Point Model with Feature Descriptors	Robust	Robust to in-plane, cannot handle large out-of-plane rotations	Robust	Robust
Geometric Shapes Model	Robust	Sensitive	Sensitive	Sensitive
Color Model	Robust	Robust	Not that Robust	Sensitive
Texture Model	Sensitive	Robust	Robust	Sensitive
Sparse Representation Model	Depends on the design of the templates	Sensitive	Handle gradual changes	Handles partial occlusion
Patch Model	Depends upon the sub-regions chosen for the appearance model	Handles gradual change	Handles gradual change	Handles partial occlusion
Active Contour Model	Depends on initialization of the contour	Handles gradual change	Robust	Sensitive
Segmentation Model	Sensitive	Sensitive	Handles gradual change	Handles partial occlusion

Hence, we can summarize the appearance models on the basis of their robustness to challenging situations like frequent changes in appearance of the target due to scale, rotation, illumination and occlusion. In Table 2.1, the above described appearance models are represented as per their response to these challenging conditions and are summarized as follows :

— The interest point model with feature descriptor proves robust in almost all the situations like scale change, illumination, occlusion and can handle gradual changes of target object's rotation.

— The geometric shape appearance model proves robust when the object does not undergo major shape change and is robust to scale change and rotation.

— The color model proves robust in case of scale change and rotation and is also very fast to compute. Since, the color histogram does not provide clear spatial information about the target object, therefore the model can sometimes produce ambiguous results in cases of occlusion. Also, the model is very sensitive to illumination change.

— Texture model proves robust in illumination and is sensitive to scale change as it depends on the size of the texture pattern. It can be used to track object with sharp

[1] In-plane rotation is considered

edges. The model can also deal with gradual rotation.

- Sparse representation proves robust in case of illumination and partial occlusion but the model fails in case of rotation as the template which is used for such model only contains information about the target object from a single viewpoint.
- Patch model can robustly deal with occlusion, illumination and rotation to a certain extent, as the target model is represented by a set of multiple patches and even if some patches become unusable due to occlusion, rotation and illumination change, the target object can still be tracked. However, the method cannot deal with scale changes of the target object.
- Active contours prove robust in illumination change and in cases where the shape of the target is undefined, as it provides detailed shape and deformation information about it. But the initialization of the contour of the target object is very difficult for this model and the rest of the tracking task depends solely on this initialization. The model is sensitive towards occlusion and can handle gradual scale and rotation changes of the target.
- Segmentation model proves robust in case of occlusion, illumination and rotation, since the pixels in the segmented region can be used for voting for the target and background regions (*backprojection*). But if the segmented regions are wrongly grouped, then the results are not reliable and this model cannot deal with scale changes.

It is evident from Table 2.1 that Interest Point with Feature Descriptors appearance model can handle tracking challenges robustly, since their method of extraction describes the target object in a dense manner by taking the neighborhood into account along with the spatial information of the keypoint feature. Further, the keypoint features can be detected by the computer easily and are repeatable, which is quite essential for feature matching and successful tracking.

2.4 Motion Model

The objective of tracking is to locate the target object in every frame of the video sequence by observing its motion. For achieving this, a motion model is used to predict how the target object is moving in a video sequence. The spatial information of the target object is taken into account for prediction of target object position in the next frame. By utilizing the inter-frame correspondences, object instances are observed and is used to predict the target object location in every frame. No matter how efficient the motion model is, the output predicted by the motion model is still an approximation and it might be possible that the target object is not located at the predicted position. Therefore, the target object is searched within a localized region around the target object's predicted location in the frame. This search region is called Region of Interest (ROI), and is often represented by a rectangle. The size of ROI varies and is chosen by experimenting with different sizes, for which the prediction results are good. But usually, the ROI is not taken very large because it is known in advance that the target object will not move very fast between two consecutive frames. In addition, a larger ROI will contain noise, possibly distractors, and will require more processing time. The most popular motion models used in object tracking are particle filter, tracking by feature matching and tracking by detection. In particle filter, spatio-temporal information about the target object is used for prediction, whereas in feature matching, the current location of the matched features are used. In tracking by detection, no spatio-temporal information is utilized for prediction of target location. The following paragraph gives a brief overview of the motion models used in visual object tracking.

2.4.1 Tracking by appearance matching

In this method, features are matched with the features present in the appearance model to obtain a similarity score that describes the percentage of how well the current tracking state (object location in the frame at a specific time) appearance matches with the appearance model. Usually, the search region for feature matching is constrained as it is assumed that the target cannot move by a large distance between two consecutive frames. However, in this method, the features can be extracted and matched in a specified search region or in the whole frame. Hence, this method gives the leverage on target searching and matching features in a wider region around the target object. Although, the location estimated of the target is a coarse estimation, but finer prediction strategies are applied to get a finer location estimation of target, as in [2].

2.4.2 Particle Filter

In this type of motion model, the problem of object tracking is formulated as a Bayesian probabilistic estimation, where it is required to predict the target object's *posterior* state (probability distribution) in the current frame, given the target object's *prior* state in the previous frames [40]. Hence, the Bayesian theory estimates the posterior probability distribution of the target object in the next frame, given the observations and the prior probability distribution up to the last state. In the particle filter framework, the posterior probability distribution of the target object is modeled by some randomly chosen particles (samples) at a certain time, instead of modeling only a Gaussian distribution. Each particle is sampled from a Gaussian distribution and assigned a weight based on its probability of observing it in the distribution. After the target object has changed its location, the probability distribution changes, and accordingly the particles are re-sampled and weighted according to the new probability estimation. The particles with lower weights are eliminated and the new particles are generated from the estimated mean value of the distribution. The advantage of this method is that it models non-linearity of the system as well as uses the temporal information obtained from the past frames, for prediction of the target object location.

2.4.3 Tracking By Detection

In this type of motion model, the target object is detected in every frame by an object detector. The detector detects the target object in each frame independently of the previous detection, and for tracking, these independent multiple detection are associated (stitched) across the frames. Hence, no spatial or temporal information is used for association of these multiple detection. Therefore, tracking by detection uses an object detector for tracking objects in the scene. Hence, it sometimes also called as sliding window approach, since the search principle of an object detector is similar, although the detection might happen only around the area of the previous frame as in a [41]. The *advantage* of using an object detector is that it helps in automatic initialization of the tracking process and also in re-initialization of tracking, in case when the target is lost. In contrast, the particle filter framework cannot be easily re-initiated. In case of abrupt change in motion of the target object, tracking by detection method can detect the target location as the method searches the whole image for possible target regions. Whereas, in case of most of the tracking methods, search for possible target region is limited in neighborhood, due to which the tracker may eventually loose the target if movements are too large [15]. The *disadvantage* of using tracking by detection approach is that the method produces false positive detection and missed detection when the appearance of the target object changes due to illumination, occlusion and rotation, which ultimately results in unreliable tracking. Whereas in tracking framework, the temporal information from the past frames helps in more reliable prediction of the target object's location. The other disadvantage is that the target object has to be known in advance in order to train the object detector or an online learning method has to be devised whereas in the tracking framework, no prior knowledge about the target object has to be known before, except its region of interest in the first frame.

2.4.4 Combination of Motion Models

As we have seen above, neither the output from particle filter or tracking by detection can be considered reliable and sufficient for tracking objects robustly in video sequences. One possible solution proposed by researchers in order to alleviate this problem and produce more reliable predictions, is to use multiple motion models for tracking of objects. In recent works, such as [42] and [15], the output from both the detector and the tracker are utilized for achieving better tracking with reduced drift errors, as compared to the tracking methods that utilize only a single motion model. In [42], the system operates in three steps. First, the target object is manually annotated and is tracked for rest of the frames and independently detection is performed for all frames by an object detector. Then, the detector output is evaluated and is updated online by a classifier. The detector also corrects the tracker in case of tracking drift and re-initialization of tracker. Through this step only, the detector and the tracker exchange information, otherwise they work independently. The advantage of their approach is that the method is robust to partial occlusions and changing poses. The output from the online classifier mutually helps in correcting the detection and in turn tracking. The disadvantage of this method is that the tracker fails if the object is occluded since the main assumption of this method that the target object has to be visible in every frame in order to track it successfully. Also, the method is unable to handle full out of plane rotation. In [15], a similar approach is applied by using both the outputs from tracking by detection and particle filter motion model, except in their method there is no exchange of information between the detector and the tracker unlike in [42]. For the final output of the system, the detection and the tracking score are evaluated and the result is given by the one that has the highest confidence score.

2.5 Face detection

As we have a prior knowledge that the target object that we are interested in tracking is a face, using a face detector can prove beneficial in improving the performance of a tracker. The advantage of using a face detector with a tracker is to re-initialize the tracker, during tracking drift or when the target remains hidden for some time due to heavy occlusion. Moreover, in case of abrupt change in motion of the target face, the face detector can locate its position, as the detector searches the whole image for possible face regions. In contrast, the tracker searches for the target location in a neighborhood. As discussed in subsection 2.4.3, we want to utilize the advantage of tracking-by-detection by using a face detector for face tracking in an interesting manner by keeping its disadvantages to the minimum. There has been significant efforts in research for face detection and an extensive literature can be

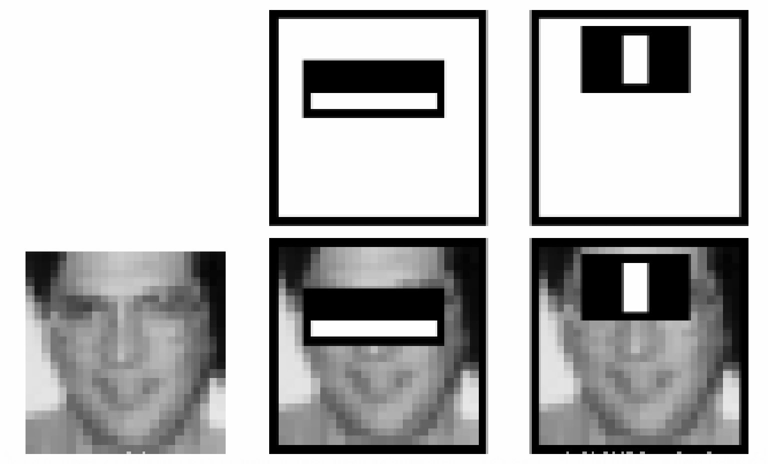


Figure 2.5 Face detection using Haar features. (This image is adapted [reprinted] from [10] © 2011 IEEE).

found in [43].

One of the pioneering work in the domain of face detection is credited to the work of Viola and Jones [10]. Their main contribution of integral image technique and Adaboost classifier learning, has made *Viola-Jones* a successful real-time face detector. Rigid Haar-feature templates, as shown in Figure 2.5, are used as face features and are detected using sliding window integral image method. These features are used for training a binary classifier based on a Adaboost type of learning method. The term "boosting" here means finding an accurate hypothesis of the face location by combining multiple weak hypothesis. Following their learning approach for face detection many face detection methods came into picture with variants of Haar-like features. The work of [44] proposed diagonal Haar-like features instead of conventional rectangle features. In [45], Haar rectangles are separated by a spatial distance to capture multiple views of the face. Lienhart et al [46], proposed an extended set of Haar features, where rotated rectangular features are used for face detection.

Apart from this, local texture features such as Local Binary Patterns (LBP) [28], MB-LBP [47], LBP Histograms [48] have been introduced for face detection. Authors in [16] use pixel differences of a face to learn a manifold. Then these features are optimally learned and combined to construct more discriminative features for detecting faces. Besides different feature representations, Lienhart et al. [49] and [50] used regression with classification and claimed to have better results than using a simple decision function. Authors in [51], proposed a part-based deformable model for tracking a non-rigid object like a face. Face detection using neural networks can be seen in [52]. In [53], the authors learn a regression face model and tried to solve the face tracking problem by incrementally learning the face model. In [54], the authors use both face detection and a landmark detector to effectively track a single face in a video sequence instead of coupling both the detection and the landmark module together as is done by Zhu et al [51]. Recently, face detection using convolutional neural networks with proposal generators are also becoming popular and giving real-time results [55].

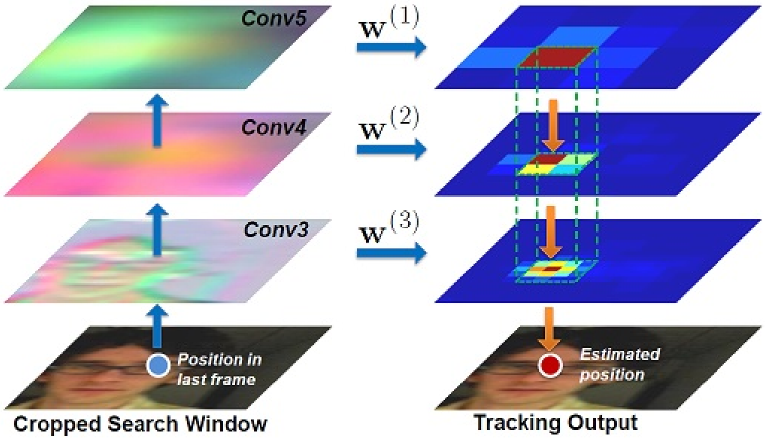


Figure 2.6 Visual Tracking using deep features in HCF. The third, fourth and fifth convolutional layers as used as target representations. The maximum value generated in a response map indicates the estimated target position. (This image is adapted [reprinted] from [11] © 2011 IEEE).

2.6 Popular state-of-the-art trackers

Table 2.2 that gives a brief overview of the different appearance and motion models used in recent state-of-the-art methods for tracking objects in video. The main challenge that an appearance model faces during tracking is to simultaneously tackle the problem of occlusion, illumination and distractors. The previous research gives an appropriate direction in solving such challenges by using visual cues from the target object and also by including some information from around the target object. The addition of contextual information is shown robust in the domain of object recognition [56]. This concept of constructing appearance can be seen in tracking methods such as [57]. Here *Supporters* are keypoint features that are spread all over the image and that have similar motion with the target object. The advantage of using supporters is to help the tracker during occlusion. Because, even if some features do not get detected during occlusion, the object location can be determined with the help of supporting features. Methods like [58], create a topology between regions that are around the target object and exploit the interaction of the regions with the target object for dealing with occlusion and distractors. Whereas in [59], the authors explore the context on-the-fly and exploit information for tracking from both *Supporters* and *Distractors*. In [60], the authors create an internal structure from features inside the target object.

Some methods decompose the target object into regions or patches and are termed as *part-based* trackers. They focus on the local features and the holistic appearance of the target object. In [61], they use a pre-defined grid and associate target region patches to the grid. But having a rigid grid may not be suitable for tracking a deformable object like a face. Hence, common aspect which can be derived from part-based methods is that decomposing the target object region into parts tackles partial occlusions.

Tracking exploiting global features such as raw intensity values as templates work by finding the most similar region by matching the appearance template. For example, the classical Mean-Shift [31] tracker uses color features to find the object of interest by comparing color histogram of the object. Trackers like [62] use subspace models to incrementally learn the object representation. Sparse representation uses a set of holistic templates to represent the target object by combining the trivial and target templates. It finds the target object by solving the L1 minimization problem. Holistic appearance models like [63], use histogram to encode object structure. Recently, hierarchical models using CNNs are becoming popular in object tracking as they have demonstrated their effectiveness in solving other visual recognition tasks [64]. HCF [11] tries to solve the problem of object tracking by using deep learning features from the hierarchical convolutional layer (refer Figure 2.6), instead of only utilizing features from the last layer of the CNN in the contrast to DLT [65].

2.7 Summary of Literature Review

Tracking is the process of estimating an object's location in a set of consecutive video frames, based on a set of observations, which is particularly distinct only to this target object during the video sequence. In all the aforementioned tracking tasks, one thing which can be found common is the appearance representation for the object for which tracking is desired. Thus, the ideal condition for robust tracking is to design an appearance model based on features that are invariant to scale, rotation, illumination, occlusion, viewpoint, etc. But, as we have discussed in Section 1.3, it does not exist yet and designing such a model is a major challenge in object tracking.

The literature summary points out that for reliable tracking of object in a video sequence, appearance and motion models form the fundamental building blocks of a tracking method. An appearance model, which gives information about the characteristics of the target object by encoding its neighborhood information, and a motion model, which dictates the search

Table 2.2 Overview of the appearance and motion models used in state-of-the-art tracking methods

Tracker	**Target Region**	**Appearance Model**	**Motion Model**
Structure Preserving Object Tracking [58]	Bounding Box	HOG gradients	Tracking By Detection
L1 [9]	Templates constructed from the initial bounding box	Sparse representation(Target + trivial template)	Particle Filter
TLD [42]	Bounding Box	Rectangular grid points inside the target region	Optical Flow
To Track or to Detect [15]	Bounding Box	Ensemble Model (RGB histogram+ Optical flow + Motion feature modeled by normal distribution)	Tracking By Detection
IVT [62]	Bounding Box	Subspace Representation template	Particle Filter
KCF (Kernelized Correlation Filters) [41]	Bounding Box	HOG templates with cyclic shifts	Tracking By Detection
CSK (Circulant Structure Kernels) [66]	Bounding Box	Raw pixel values with cyclic shifts	Tracking By Detection
Adaptive color attributes for real-time visual tracking [67]	Bounding Box	Color features with cyclic shifts	Tracking By Detection
Highly Nonrigid Object Tracking via Patch-Based Dynamic Appearance Modeling [34]	Patches sampled from Bounding Box	HSV histogram	Particle Filter
Structure-Aware Keypoint Tracking for Partial Occlusion Handling [60]	Circle	RGB Histogram + SIFT keypoints	Particle Filter + keypoint matching
Robust Object Tracking with Online Multiple Instance Learning [5]	Patches sampled from Bounding Box	Haar gradients	Tracking By Detection
Learning a Deep Compact Image Representation for Visual Tracking [65]	Bounding Box	Convolutional layers (offline trained)	Particle Filter
Hierarchical convolutional features for visual tracking [11]	Bounding Box	Hierarchical Convolutional layers (offline trained)	Sliding Window

region for locating the target object taking into account the spatio-temporal information about the target object, is a must requirement for reliable tracking in a video sequence.

Further, in Table 2.2, the majority of the trackers are initialized by a bounding box. The bounding boxes are manually annotated around the target object, or are obtained by using an object detector, which usually outputs bounding boxes for detected objects in a frame. Most of the time, the initialization is manual. This is because it is very challenging to localize an object of interest to track automatically.

Color RGB or HSV histogram is the most widely used appearance model for object tracking as it is robust to tracking challenges such as change in orientation, relative position and occlusion, and are simple to interpret [31]. For motion model in tracking, it is noted that tracking-by-detection is most commonly used in the recent times. Furthermore, an object detector can facilitate tracking by injecting prior knowledge about the target object that is required to be tracked and re-initialize the tracker during tracking drift.

Through the extensive literature survey, we realized that tracking challenges such as low-resolution of video sequences and tracking an object in a lengthy video sequence, are often under represented in the existing evaluation experiments. Hence, providing an opportunity for the future researchers to explore this direction.

CHAPTER 3 OVERVIEW

This thesis proposes solutions for the challenges associated with tracking an object of interest (any non-rigid object, for example like a face) accurately, and for a long duration of time in a video sequence in the presence of distractors. Further, the object might get deformed during online tracking and may disappear from the sequence for some time as well and again reappear in later part of the sequence. Finally, it is not always necessary that the video sequence are always captured in high resolution, and motion of the object is not constrained between two consecutive frames. Considering the aforementioned challenges, the proposed solutions are summarized in three contributions that are presented in Chapters 4, 5 and 6.

3.1 Progression of Research

1. The first article, CTSE [1], proposes a solution to solve the problem of accurately estimating the object location and tackling the influence of similar looking objects in the environment so as to prevent the tracker from getting confused. In this solution, the appearance model of the object for tracking is based on keypoints that are voting for the object center location. Since it is not always possible to have a large amount of samples for a face of interest, in this article, tracking is estimated as *model-free tracking*, where the only information provided to the tracker is from an initialized bounding box in a single image frame. To maximize the information available to the tracker, *contextual* information is also provided to the tracker. Not any context information, only the context that has *correlated* motion with the center of the object. The tracking features are evaluated for their quality online during the tracking process and the latent features are discarded from the appearance model. In order to localize the object correctly, a novel strategy is proposed to deal with the inherent tracking noise during the process of tracking in video sequences, making it performing at the top on challenging video dataset.
2. As the research progressed, it became evident that for a non-rigid deformable object, it is necessary to take into account multiple features for tracking. Hence, the exploration of features that are robust to the global and local appearance changes, combined with the aforementioned model-free tracker came into realization. Therefore, the second article, TUNA [2] tries to solve the problem of deformation and scale change of the target object by using multiple appearance models. Further, a robust matching criteria is proposed to tackle the fast motion of the target object. Moreover, a strategy is

proposed to add new features and delete non-performing features to the appearance model, which helps in tracking the target object over long periods in video sequences.

3. Finally, the *third* article is presented in Chapter 6. It proposes a novel Face Tracking system called *FaceTrack* that utilizes the advantage of tracking by detection by incorporating a face detector [3]. It has been observed that neither detection nor tracking *alone* can solve the complex challenges that occur while tracking faces. The advantage of using a face detector is that it provides prior information about the target face and can handle abrupt motion and scale changes of the target face, but cannot handle occlusions very well. On the other hand, the advantage of using a face tracker with multiple appearance models address the problem of face appearance matching in real-world unconstrained situations. For better localization of the face tracking output, face candidates are generated that are computed for appearance similarity scores. Finally, a weighted score-level strategy is proposed for selecting the best face candidate as face tracking output.

3.2 Experimentation

For each of the research milestones, extensive experimentation was performed on the visual object tracking evaluation benchmark [68]. The experimental results were reproduced multiple times and thus ensuring that our observations are significant. Appropriate representation for explaining the tracking system is done by using block diagrams. And precision and success curves are used to represent the tracking results. These have been discussed in detail in subsequent chapters.

CHAPTER 4 ARTICLE 1 : CONTEXTUAL OBJECT TRACKER WITH STRUCTURE ENCODING

Authors

Tanushri Chakravorty, Guillaume-Alexandre Bilodeau,
LITIV Lab., Polytechnique Montréal
Eric Granger, *LIVIA, École de technologie supérieure, Montréal*

E-mail : {tanushri.chakravorty, gabilodeau}@polymtl.ca, eric.granger@etsmtl.ca

Published in International Conference on Image Processing, ICIP, December, 2015

Reference as T. Chakravorty, G. A. Bilodeau and E. Granger, "Contextual object tracker with structure encoding," *2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, 2015, pp. 4937-4941.*

4.1 Abstract

Motivated by the problem of object tracking in video sequences, this paper presents a new Contextual Object Tracker with Structural Encoding (CTSE). The novelty in our tracking approach lies in the application of contextual and structural information (that is specific to a target object) into a *model-free* tracker. This is first achieved by including features from a *complementary* region having correlated motion with the target object. Second, a *local structure* that represents a spatial constraint between features within the target object are included. SIFT keypoints are used as features to encode both these information. The tracking is done in three steps. Firstly, keypoints are detected and described to encode object structure. Secondly, they are matched in every frame. Finally, each matched keypoint votes for the target object location locally in a *voting matrix* by using the encoded object structure. The voting method gives more priority to the keypoints that have been matched more often and are closest to the target's center than the rest. The proposed tracker is competitive with state-of-the art trackers while being significantly faster. It ranks as first or second most accurate tracker in experiments with standard datasets.

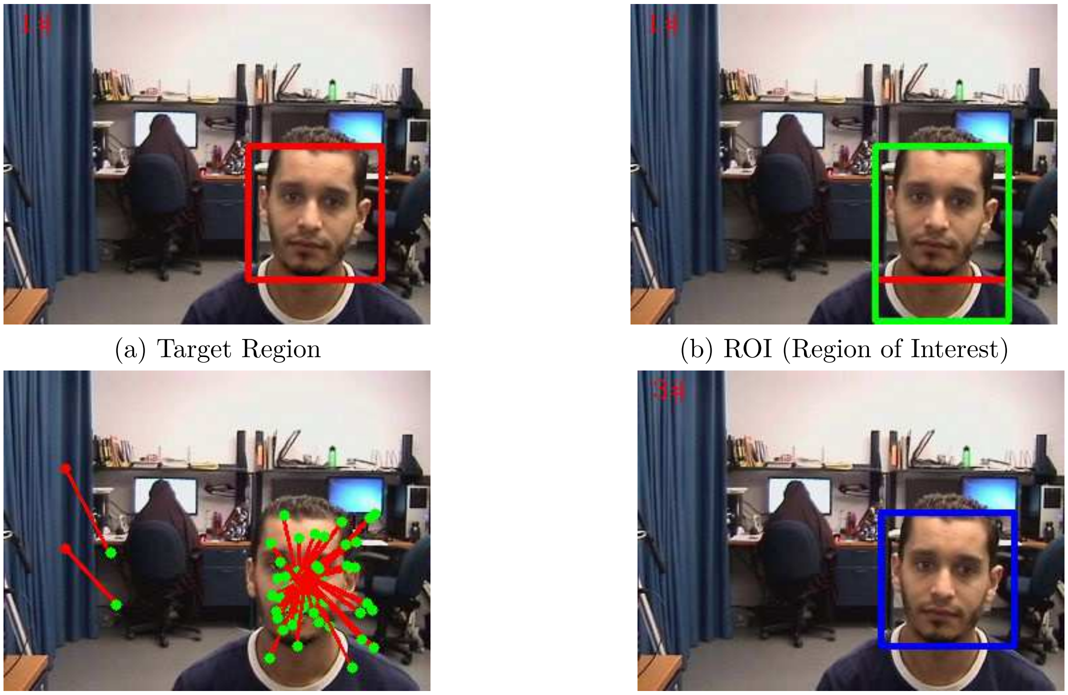
Index Terms- *Object tracking, Model-free tracking, Context, Appearance model, Object structure, Keypoints*

4.2 Introduction

Even after decades of research, object tracking in a real-world unconstrained environment remains an arduous task. The core problem in any tracking algorithm occurs due to abrupt and frequent appearance changes of the target object because of illumination, occlusion, scale and presence of objects having similar appearance to the target object (distractors) in the environment. Several approaches have been proposed to design strong appearance model, in order to discriminate an object from the background and match that appearance model in every frame, so as to have strong similarity measure for accurate tracking. Still, it is difficult to simultaneously address the problem of appearance changes caused by occlusion and illumination, and the problem of distractors. This paper presents a tracking method that can jointly address these problems.

The previous research gives an appropriate direction to solve these problems by focusing the target's appearance model – not only on the object's region description but also on the visual cues around the target object. The addition of contextual information besides the target object's region has been shown successful in the domain of object recognition [56] and semantic segmentation [69]. Approaches like [58], [70] explore the use of context in tracking. Both methods create a structure (topology) between correlated regions (having similar motion) and the main target object, and exploit this structure for object tracking. In recent work, the idea of using contextual information is slightly different. In [57], they use *Supporters* which are keypoint features spread over all the image and not necessarily around the target object that bear a correlated motion with the target. In [60], instead of using a set of image features from the whole frame, authors use features from the target itself and create an internal structure for all such features. The first common aspect is the use of contextual information. It is the data available from or around the target object having a correlated motion with the target object. The second common aspect is to use the structural information between the target object and the correlated features for efficient object tracking, thus dealing with distractors.

Another inspiration for our tracker comes from part-based trackers [61], [71], [63], where the target object is described by decomposing the object's region description into parts or patches. In [61], they use generative representation that belong to the target object only with patches pre-defined in a grid. These patches vote for the target object position in a competitive approach. However, their method becomes inappropriate for tracking non-rigid objects as the grid is unable to adjust to changes that occur due to deformations. In [71], they sample a set of overlapped patches and track object using visible patches during partial occlusion. In [63], they propose to use a histogram based model to encode the object structure. Part-based



(a) Target Region

(b) ROI (Region of Interest)

(c) Voting by keypoints

(d) Final Output

Figure 4.1 Proposed Tracking Method (a) Target Region (b) ROI (Target + Complementary Region (below the red line)) (c) Voting by keypoints (green dots) for the target location (red dot) (d) Final Output by tracker

trackers like [72] and model-free trackers like [5], [42], [73], use discriminative representation and learning approaches to distinguish the target object from the background. From this, the common aspect is that decomposing the object's region is robust to partial occlusions.

This paper presents a new model-based tracker entitled the Contextual Object Tracker with Structural Encoding (CTSE). It takes as input all the information about the target object and its context from the first frame and then tries to locate and update this pattern of input correctly for the rest of the video sequences. The CTSE is illustrated in Figure 4.1 and follows a three step process [1]. First, SIFT keypoints are extracted and described for the ROI (target + complementary). They provide invariance to illumination and robustness against distractors. The local structure for the target region is also computed. This provides the robustness against occlusion because the location of the target is described uniquely with respect to each individual keypoints. Thus, each keypoint behaves as a part of the object's region description. Second, these keypoints are matched, and each matching keypoint votes individually for the target position in a voting matrix. Third, in locations where multiple votes form a cluster, the global maximum of the obtained votes is selected as the final target location. Finally, the model is updated. Some key contributions to the state-of-the-art are as follows :

1. Keypoints having a *structure* spatial constraint and a *motion* correlation with the target center, are shown to be robust features for object tracking. Hence, principles from both context and structure may be combined into object tracking.
2. To achieve greater tracking accuracy, the inherent noise of the tracking method is modulated by utilizing a technique called *voting by keypoints*. In this, the voting for the target location is done using the structural configuration of each feature (keypoint).
3. The *quality* for every keypoint feature is estimated by maintaining a *structural configuration* for each keypoint. This helps in achieving a finer global prediction for the target location in every frame. The structural configuration is updated on the fly so as to accommodate the appearance changes.

Section 2 describes our appearance model and the tracking method. Section 3 describes the update steps for the structural configuration for a keypoint. Section 4 includes experimental results, and Section 5 draws conclusions. Results of the proposed tracker are compared to reference trackers using the video sequences in [68], and [60] respectively.

[1] https ://bitbucket.org/tanushri/ctse

4.3 Tracking Method

Generally, a tracking algorithm includes two main components : (1) appearance model that represents the characteristics of the target object, and (2) a search strategy to estimate the target's position in every frame.

4.3.1 Appearance model (target and complementary region)

Considering the underlying concept of model-free tracking, our tracker is initialized in the first frame by annotating a bounding box and a complementary region for a target face as shown in Figure 4.1b. Using this as our region of interest, SIFT keypoints are first detected all over the frame and then the keypoints contained inside the ROI are stored. Consider a target with a set of keypoints in the appearance model stored in a vector K. Let each keypoint be denoted by k_i, such that $k_i \in K$. We used SIFT keypoints as literature has shown that they are invariant to scale, translation, illumination and can handle small rotation variation, which makes it a very suitable feature for object tracking [74].

Encoding structure

As soon as the keypoints are detected and their descriptors are computed, a *structural* configuration for each keypoint is initialized. The structural configuration is represented as $S_{k_i} = [d_{k_i}, X_{k_i}, C_{k_i}, p_{k_i}]$ and consists of the following :

1. d_{k_i} = descriptor of keypoint
2. $X_{k_i}[\Delta x, \Delta y]$ (Spatial Constraint Vector) = Describes the keypoints location with respect to the target center.
3. C_{k_i} (Correlation Factor) = Indicates the keypoint's motion correlation with respect to the target center.
4. p_{k_i} (Proximity Factor) = Describes the importance of the keypoint's proximity to the target center. A keypoint located nearby to the target center will have higher proximity value as compared to others. It has a direct effect on the C_{k_i} parameter of keypoint configuration as we will see in later subsections.

The encoded structure with spatial constraints helps in predicting the target's position in the next frame as the structure will remain mostly unchanged for the future frames of the video sequences. Therefore, when the target moves in the next frame, the points that have a correlated motion with the target center will also move by the same spatial translation in the next frame, but the relative distance (spatial constraint vector, X_{k_i}) of these points from the center will be constant. Hence, by re-detecting and matching the same keypoints as present in the appearance model for a target in the next frame t+1, we can estimate the new position of the target. The advantage of using such a structure aids in tracking during occlusion because even if a single keypoint is matched during occlusion (as rest of the keypoints will be hidden), the target object can still be tracked.

4.3.2 Search strategy

First SIFT keypoints are detected and described in the whole frame. These are matched with those present in the appearance model by comparing the Euclidean distance similarity between their descriptors. The advantage of detecting the keypoints in the whole frame helps in matching keypoints with the appearance model even if the target undergoes large or abrupt motion. The keypoint matching outputs a region with keypoints that co-occur with those present in the appearance model. We use a similar criteria as used [22] for removing erroneous matches and keep only those matches that have a distance ratio less than 0.8. The matching output gives a region, which is a coarse estimation of the target. Therefore, for finer prediction of the target location we have to use a different strategy called *voting by keypoints*.

Voting by keypoints using encoding structure

During tracking, there is inherent noise of the system, which will influence the target's center prediction by each keypoint. Therefore, we consider this inherent noise while estimating the final target location. We assume that all the pixels in the frame are affected by the same inherent noise and associate a single Gaussian pdf (probability density function) to all k_i. We want to vote in such a manner that a pixel on a patch around k_i, will have the highest vote with its closeness to the patch's center indicated by k_i (similar to a Gaussian function). Thus, k_i votes for the target's center by using its X_{k_i} parameter of S_{k_i}. Lets say the current position of k_i is x in the frame t and its corresponding structural spatial constraint is denoted by X_{k_i}. Hence, the Gaussian pdf with which k_i will cast its vote can be written as :

$$P(x|k_i) \propto \frac{1}{\sqrt{2\pi|\Sigma|}} exp(-0.5(x-X)^T \Sigma^{-1}(x-X)) \tag{4.1}$$

Here, Σ is a covariance matrix. Therefore, the local prediction by given k_i for the target's new center location is given by :

$$x_{LocPred_{k_i}} = P(x \mid k_i)_{k_i} \mathbb{1}_{(k_i \in K)} \tag{4.2}$$

Hence, each keypoint *votes* for the target's center location with a Gaussian pdf and its Correlation factor C_{k_i} and $\mathbb{1}_{(k^{(i)} \in K)}$ is an indicator function, which is set to one for keypoints that are matched in current frame. All such individual votes are summarized in a vote matrix. In order to select the most probable location of the target center, we find the location inside the vote matrix (VM) where the sum of individual votes is highest, resulting in a cluster of votes. This shows that a cluster of keypoints have voted for the same center location for a target object. Hence, the final target center is given by Equation 4.3, and is represented as follows :

$$x_{targetCenter} = \arg\max_{x \in VM}(\sum_{i=0}^{K}(P(x|k_i)C_{k_i}\mathbb{1}_{(k_i \in K)})) \tag{4.3}$$

4.4 Determining Keypoint Quality

4.4.1 Adaptive correlation and proximity factor

As seen from Equation 4.2, the correlation factor C_{k_i} plays a major contribution in determining the global prediction for the target center. Initially all the keypoints in K are assigned with an initial value for C_{k_i}. With every new frame processed t, the C_{k_i} parameter value in the structural configuration of keypoint k_i updates with *learning factor*, α using Equation 4.4 as follows :

$$C_{k_i}^{t+1} = (1-\alpha)C_{k_i}^{t} + \alpha p_{k_i}^{t}\mathbb{1}_{(k_i \in K)} \tag{4.4}$$

Here the term $p_{k_i}^t$ represents the proximity factor for a particular keypoint, k_i at frame t. The p_{k_i} for a particular varies non-linearly with its closeness to the target's center and is evaluated by using a function, given by the following Equation 4.5 :

$$p_{k_i}^t = max((1 - |\lambda(x_{TargetCenter} - x_{LocPred_{k_i}})|), 0.0) \tag{4.5}$$

Here λ is a constant. Hence, a keypoint that is closer to the predicted target center ($x_{TargetCenter}$), will have more importance in contributing its vote for in the Gaussian pdf (Refer Equation 4.2) in the next frame $t+1$, than those which are far from the target center. By doing this, we achieve higher accuracy for target center location because even if certain keypoints that are erroneously matched, they will have a very less contribution in vote matrix. For the rest of the keypoints that have not been matched, their C_{k_i} reduces (Refer Equation 4.4).

Table 4.1 Comparison of CLE and OR of proposed tracker with respect to state-of-the-art part-based trackers. **Bold** red indicates the best results and blue italics indicates the second best.

Videos	**SPT**[72]		**SCMT**[63]		**AST**[71]		**SAT**[60]		**CTSE**(proposed)		**CTSE**(no context)	
	CLE	OR	CLE	OR	CLE	OR	CLE	OR	CLE	OR	CLE	OR
FaceOcc1	116.84	0.05	5.07	**1.00**	85.43	0.25	14.26	*0.99*	**3.77**	**1.00**	*3.89*	**1.00**
Girl	**8.97**	**0.84**	201.27	0.19	53.42	0.17	*10.01*	**0.84**	10.52	*0.78*	10.61	*0.78*
David	36.09	*0.62*	33.81	0.60	68.57	0.37	**10.48**	**1.00**	*26.38*	0.60	*26.38*	0.60
Cliff bar	*22.11*	0.51	77.31	0.24	35.35	**0.69**	25.33	*0.60*	26.13	0.51	**20.68**	0.59
jp1	35.21	0.18	17.74	0.78	16.66	0.84	*7.03*	*0.89*	**5.95**	**0.99**	**5.95**	**0.99**
jp2	30.58	0.39	69.44	0.55	45.15	0.55	*7.25*	*0.93*	**3.91**	**0.99**	**3.91**	**0.99**
wdesk	79.92	0.13	34.17	0.57	80.97	0.32	**11.12**	**0.90**	*11.23*	*0.85*	*11.23*	*0.85*
wbook	11.27	0.98	**5.09**	**1.00**	8.68	*0.99*	11.87	*0.99*	*6.92*	*0.99*	*6.92*	*0.99*

4.5 Experimental Results

For comparison, we use state-of-the-art evaluation criteria namely, bounding box *Overlap Ratio* (**OR**) and average *Center Location Error* (**CLE**). OR is the average percentage of frames where the overlap of BB's (bounding boxes) of tracker and ground truth is at least 50%. CLE is the Euclidean distance between the center's of BB's of tracking output and ground truth. The videos for validation have the following attributes : partial and long term occlusion (*FaceOcc1*, [61], *wbook and wdesk*, [60]), illumination, large camera motion and background change (*David*, [5], *Girl*, [6]), Background Clutter (*Cliffbar*[5]), and moderately crowded scene (*jp1, jp2* [60]). As seen from Table 4.1, the performance of our method is very good for scenes with distractors, *jp1*, *jp2*. Our method delivers a precision of 0.99 with the least error as compared to rest of the trackers, because the encoded structure and keypoints prevents the tracker from switching to distractors. The voting by keypoints using the structure helps in greatly reducing the error. The encoded structure with the complementary region helps in prediction of target during long-term partial occlusion with a precision of 1.00 in *FaceOcc1*, 0.85 in *wdesk*, and 0.99 in *wbook*, as the subset of features help in target prediction even when a significant part of the target object remains hidden for a long time. For *David* and *Girl*, our method gives a competitive performance with the ability to track objects during large camera motion and illumination change. When experimented our tracker without a complementary region (no context), the error slightly increases for *FaceOcc1*, and *Girl*, as the target region remains hidden for sometime. This results in lesser matching, and lesser impact of correlation factor in voting. Whereas for *Cliffbar*, the results significantly improve without the context, as the complementary region takes into account the background region with no motion correlation with the target, which is otherwise observed in videos having torso and head (*FaceOcc1,Girl*). Note, if both (head+torso) are occluded at the same time, the context is less advantageous, and thus being scenario dependent. Hence, context is useful for videos where other objects have correlated motion with the target. For a typical 320x240 resolution video sequence, our tracker runs with 10 frames per second on Intel Core i7, 3.40 GHz machine. Figure 4.2 shows the qualitative results of our tracking method.

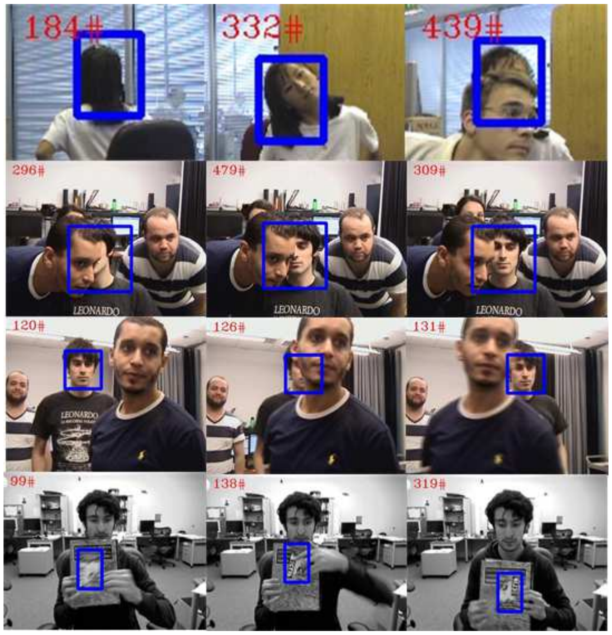


Figure 4.2 Qualitative Results. From left to right row-wise and top to bottom : *Girl*, *jp1*, *jp2*, *Cliffbar* video sequences.

4.6 Conclusion

In this paper, a new tracker has been proposed that combines the concept of context and structure for object tracking. Experimental results have shown that using keypoint features that have a correlated motion with the target center and that are organized in a structure having spatial constraints with respect to the target center, are robust features for object tracking in video sequences. Our results emphasize that by adapting the structural configuration parameters of the keypoints, improves tracking for challenges such as partial and long-term occlusion, illumination, and distractors, etc. However, the robustness of our tracker depends on the keypoint detector. The future research will seek to increase the precision of our tracker by combining a detector with our method.

Acknowledgment This work was supported by fonds de recherche du Québec - Nature et technologies Team research project grant #167442.

CHAPTER 5 ARTICLE 2 : TRACKING USING NUMEROUS ANCHOR POINTS

Authors

Tanushri Chakravorty, Guillaume-Alexandre Bilodeau,
LITIV Lab., Polytechnique Montréal
Eric Granger, *LIVIA, École de technologie supérieure, Montréal*

E-mail : {tanushri.chakravorty, gabilodeau}@polymtl.ca, eric.granger@etsmtl.ca

Submitted to Machine Vision and Applications, MVA, 14, July, 2016

Reference as T. Chakravorty, G. A. Bilodeau and E. Granger, "Tracking using Numerous Anchor points," *arXiv preprint arXiv :1702.02012*

5.1 Abstract

In this paper, an online adaptive model-free tracker is proposed to track single objects in video sequences to deal with real-world tracking challenges like low-resolution, object deformation, occlusion and motion blur. The novelty lies in the construction of a strong appearance model that captures features from the initialized bounding box and then are assembled into anchor-point features. These features memorize the global pattern of the object and have an internal star graph-like structure. These features are unique and flexible and helps tracking generic and deformable objects with no limitation on specific objects. In addition, the relevance of each feature is evaluated online using short-term consistency and long-term consistency. These parameters are adapted to retain consistent features that vote for the object location and that deal with outliers for long-term tracking scenarios. Additionally, voting in a Gaussian manner helps in tackling inherent noise of the tracking system and in accurate object localization. Furthermore, the proposed tracker uses pairwise distance measure to cope with scale variations and combines pixel-level binary features and global weighted color features for model update. Finally, experimental results on a visual tracking benchmark dataset are presented to demonstrate the effectiveness and competitiveness of the proposed tracker.

Keywords Visual Object Tracking, Keypoints, Star-like structure, Gaussian, voting, model-free tracker

5.2 Introduction

Visual object tracking can be considered as a task of detecting and locating an object of interest in a given video sequence. The object may undergo appearance variations due to illumination, occlusions, deformations, motion blur, etc. Also, the presence of similar looking objects (distractors) in the scene makes the tracking task more arduous. Despite the abundance of research on object tracking in the computer vision literature, there is no available full stack tracker that can address wide-range tracking challenges. Thus, there lies a scope of improvement for developing more accurate visual object trackers.

Domain specific applications like face tracking, human, pedestrian or hand tracking allows the algorithm designer to make some prior assumptions about the appearance of the object. Although well suited for some specific applications, they target specific objects. A tracker that can be generalized to a variety of objects is often more desirable. Therefore, building on such concepts lead to the notion of model-free trackers [73]. In such trackers, the initialization is performed in the first frame using a bounding box, and the sole information on the object to be tracked is derived from that first frame. Our proposed approach is a model-free tracker, where the initialization is performed using an axis-aligned bounding box.

In order to track an object efficiently, three aspects are crucial for any tracking process. First, building an appearance model that describes unique cues of the object such that it can be detected and tracked. Hence, the appearance model must consist of strong features that provides *evidence* of an object's presence. The appearance model should be *flexible* for tackling appearance variations of the object. Finally, the appearance model should be updated at the *correct* time so as to accommodate environmental changes due to illumination, scale, orientation etc. Therefore, a correct updating technique has to be determined to prevent erroneous features from being included in the appearance model.

In the proposed tracker, these three crucial aspects are considered, including a fourth crucial aspect related to the third i.e., preserving *consistent* features in the appearance model for object localization, while removing *inconsistent* features. In our tracker, the short-term and long-term consistency of a feature is evaluated at every frame during the tracking process. Together, the long and short-term consistencies help to predict stable outputs and prevent the tracker from becoming overly sensitive to sudden changes in the environment. Moreover, including this fourth aspect also helps to track object in long-term tracking sequences, since the consistent features are retained in the model to locate the object.

The rich representations and feature models provided by deep learning methods [75], [65] are growing popular for visual object tracking and are delivering state-of-the-art results, however, they incur higher computational cost, which is highly undesirable for tracking applications. On the other hand, simpler models based on color features and keypoints are capable of capturing distinct cues of the object, and perform equally well or sometimes even better than rich models in some scenarios [76]. Therefore, thinking along those lines, we propose an *anchor-point* appearance model. Numerous keypoints on the object serve as anchor points, and are arranged in a structure defined with respect to the object center. Each keypoint predicts the object center location with its respective structure acting as anchor for the object center prediction. This structure of keypoints encoded with the object center helps to deal with occlusion and object deformation tracking challenges.

For deducing an accurate update strategy for a tracker, we believe that it is important to take advantage of both local and global features of the object. With the advent of binary feature descriptors like BRISK [30] and FREAK [27], it has become possible to find similar regions in an image at a lower computational cost. But, as they process larger image regions, it is difficult to identify local appearance changes at the pixel level. The LBSP (Local Binary Similarity Pattern) [77] binary descriptor provides pixel level change detection. Instead of comparing patches, comparisons are done at the pixel level. To identify appearance changes at the global level, RGB color information is used. Taken together, binary descriptors and color information help in successful update of the appearance model because they prevent unwanted update at wrong time during the tracking process, for exampled during an occlusion, which might result in tracker drifts and track loss.

For accurate object localization, it is important to take account of inherent pixel noise of the tracking process. Particle-filter based methods like [60], [9], and motion-based methods like [78], are classic approaches for object localization, but do not consider the inherent pixel noise caused by local deformations during tracking. Hence, a Gaussian prediction strategy proposed by [1] can be utilized to deal with the above stated challenge as it helps to compensate for the keypoint feature displacement during scale change and fast motion of the object.

Finally, it is important to retain robust discriminant features in the appearance model for a tracker to be successful. Hence, an online method should be devised to determine the consistency of features during tracking. Thus, in our proposed tracker, for each feature, a consistency (long and short-term) is determined. The long-term consistency helps to retain consistent features for tracking and short-term consistency helps to control the sensitivity of the tracker to sudden appearance changes due to occlusion, illumination variation etc. Hence, consistent features should be kept in the model and others should be removed quickly or ignored temporarily.

The contributions of this paper can be summarized as follows. First, a new model-free tracker called TUNA (Tracking Using Numerous Anchor points) is proposed to track generic objects, with a novel appearance model that captures local and the global information about an object. This information is captured using numerous keypoint features that are assembled into *anchor* points. They record the global structure of the object with respect to its center and the local information with its keypoint descriptor. Unlike other appearance models that emphasize on a single type of representation (either local or global), our model encapsulates both local and global features for a robust representation of an object. This new representation is distinctive and helps in dealing with distractors present in the environment. Second, a new updating strategy for appearance model is proposed using a combination of pixel level binary features and global level color features that determines the appropriate time for the anchor-point appearance model update. Third, a novel technique is proposed for determining scale changes. Unlike other methods [79], [80], where transformation matrices are initially computed for adjusting scale, we propose a pairwise distance method between keypoint features for estimating scale change of the object. Finally, to preserve robust features for tracking, a long and short-term consistency of a feature is estimated and evaluated online during tracking. The long-term consistency aids in retaining consistent features for tracking and evolves (increase and decrease accordingly) with the tracking process, whereas the short-term consistency is evaluated instantaneously and aids in controlling the sensitivity of the tracker to sudden appearance changes due to occlusion, illumination variations, etc. Additionally, a strategy to deal with object deformation and occlusion is proposed with a Gaussian voting for accurate object localization.

The remaining of the paper is organized as follows. Section 5.3 describes the related research work in visual object tracking. Section 5.4 describes describes the concept of the proposed appearance model. Section 5.5 and 5.6 describes the tracking framework. Experimental results and analysis are presented in Section 5.7. Finally Section 5.8 draws the conclusions.

5.3 Related Work

In this section, different representations used by trackers to model the appearance of objects are presented. Generally, object representations can be classified into two broad categories, i.e. generative and discriminative. In generative representations, the object is modeled using features extracted from the object and then the object is matched by finding the most similar region compared to the model as in template matching trackers [9], [18]. For example, Mean-Shift tracker [31] uses color features to find the object of interest, and Frag-Track [61] models the object using histograms of local patches. Trackers like IVT [62] use subspace models to incrementally learn the object representation. Sparse representation trackers like [81], [82], consider a set of linear combination of templates to represent the object.

In contrast, discriminative representations consider tracking as a binary classification task. CSK [66] uses color features and employs an online binary classifier for tracking. OAB [73] updates discriminative features via online boosting methods. Struck [79] uses an SVM classifier to generate and learn the labels online for tracking and KCF [41] samples the region around the target. The cyclic shifts simulates translations of the target object. TLD [42] uses two types of experts to train the detector online while tracking.

Part-based trackers [63], [71] divide the object in smaller regions or patches, while [72] uses superpixels as discriminative features and use learning to distinguish the object from background. The work of Cai et al. [83], proposes to decompose the object into superpixels and then use graph matching to find the association among frames.

Some trackers like CAT [70] and SemiT [84], use contextual information or supporting regions to deal with occlusion. Some authors combine multiple features [85], or multiple trackers [86], to maintain multiple appearance models. An extensive summary on various appearance model representations and visual object trackers can be found in [19] and [21] respectively.

More related to our work are Keypoint-Based trackers SAT [60], CMT [80], and CTSE [1]. SAT [60] uses a circular region for initializing tracking and computes a color histogram for that region. Further, keypoints are detected for the same region. For limiting the search region it uses a particle filter framework for keypoint detection and matching for the next frame. It uses a histogram filtering method for estimating the quality of tracking. CMT [80] uses optical flow and consensus method that aids in finding reliable matches and hence improve tracking but do not perform appearance update of the keypoint model. CTSE [1] uses a structural configuration of keypoint features to track an object and refrain from updating the model. In contrast to previous keypoint-based tracking algorithms where the search region is limited, our proposed tracker searches the entire image for finding matches and cross checks these matches for mutual correspondence for higher reliability. This way our tracker can track object that have fast motion. The proposed appearance star graph-like model tackles object deformation due to appearance change of the object. Our method also introduces the concept of short and long-term consistency of a keypoint feature. Together, the consistencies helps to retain good features in the appearance model for object location and predict stable outputs for object location by temporarily ignoring some keypoints present in the anchor point appearance model, yet keeping them in the model if they usually predict well.

5.4 Ideation

The model is inspired by deformable parts that has been used in the domain of object recognition and detection [17]. In their method, the object is divided into smaller parts that are arranged in a star graph-like configuration. Each part is represented directly or indirectly in terms of other parts, and thus there is interaction among them. In our approach, the idea of interaction is slightly different. Here, the keypoints are described in relation to the center of the object by a vector (See Figure 5.1). Thus, the keypoints are expressed in relation to the object center and not in terms of each other and thus can be considered as anchor points. Hence, except for object center position, no information is shared among the keypoints, which are unique and independent from each other.

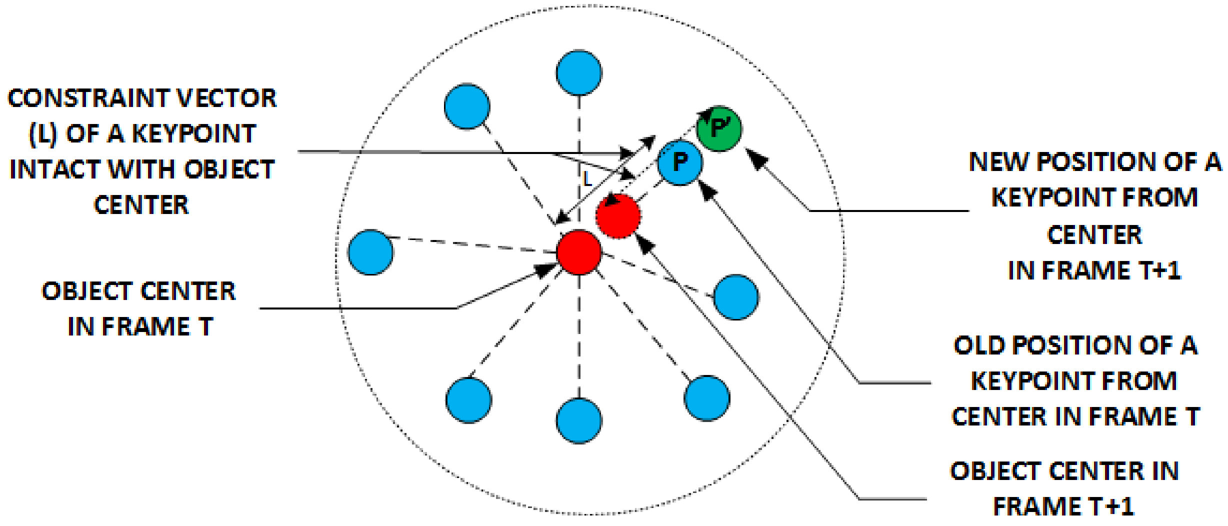

CONSTRAINT VECTOR (L) OF A KEYPOINT INTACT WITH OBJECT CENTER
L
P'
P
OBJECT CENTER IN FRAME T
NEW POSITION OF A KEYPOINT FROM CENTER IN FRAME T+1
OLD POSITION OF A KEYPOINT FROM CENTER IN FRAME T
OBJECT CENTER IN FRAME T+1

Figure 5.1 Anchor point Appearance Model. Note when the object moves, blue keypoint shifts to green (P to P'), its position changes but the encoded constrained vector L is intact.

The interaction of keypoints with the center of the object is quite unique, as these keypoints belonging to the object bear a similar motion with respect to the object center. Our hypothesis is that keypoints with a constrained vector structure that have similar motion with respect to object center helps in predicting object's position in the next frame, because this encoded structure represents a strong feature of the object that has been already learnt with the help of anchor-point features (See Figure 5.1). Therefore, when the object moves in the next frame, the keypoints with respect to the object center will also move by the same spatial translation, keeping the constrained vector of these keypoints approximately constant with the new position (P') of the re-matched keypoint as the reference. Hence, by re-detecting and matching the same keypoints for an object in the next frame, the new object position can be located.

This model is robust to heavy occlusion, as independent acting keypoints can be detected and tracked even if some keypoints become latent (not visible) during the tracking process. Our novel appearance model is efficient for tackling tracking challenges like distractors, occlusions (long and short), illumination variations, because the keypoints with their structured vector point to the object center to locate it and vote with their short-term and long-term consistencies. The long-term consistency is adapted online for a keypoint feature and aids in retaining good learnt keypoint features in the anchor-point appearance model, whereas the short-term consistency is an evaluation of a prediction response by a keypoint feature for current frame. Therefore, even if some keypoints become latent, still the location can be predicted using other visible keypoints. The short-term and long-term consistencies associated with a keypoint act as a feature learning memory. The voting by an anchor-point for the object center is performed using a gaussian window, which compensates for the keypoint displacement during object deformation. Further, the constrained vector is distinctive and tackles with distractors and background. Finally, the proposed model is not limited to specific objects and thus can be applied to a wide range of embedded vision robotics and surveillance applications.

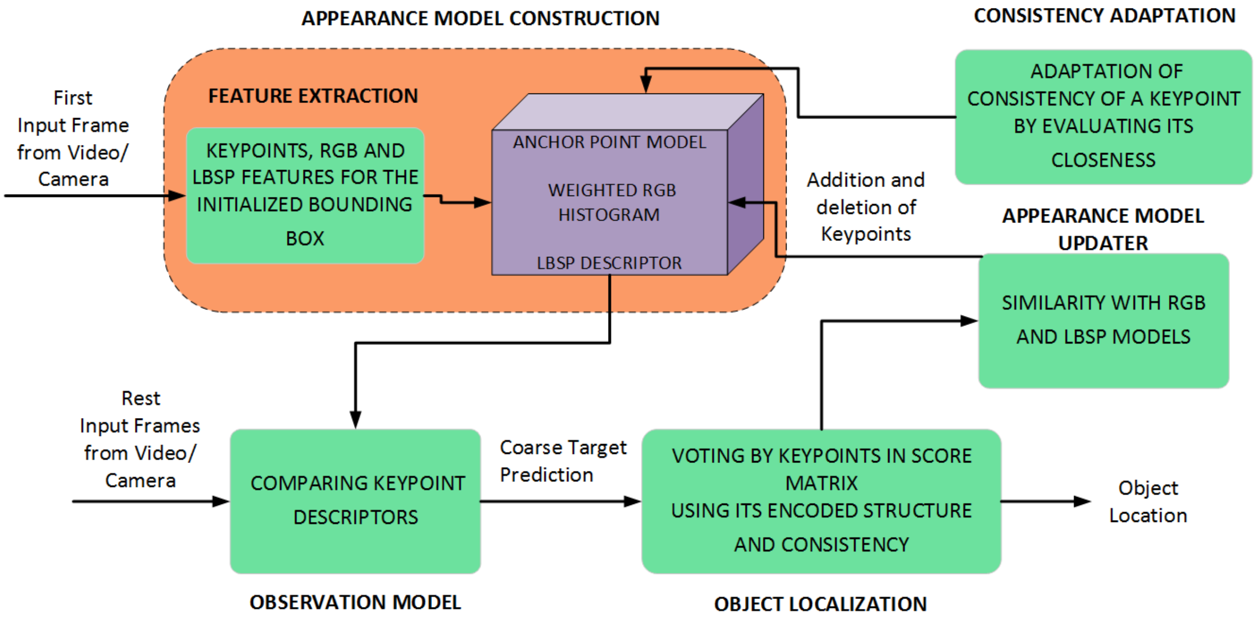

APPEARANCE MODEL CONSTRUCTION
CONSISTENCY ADAPTATION
FEATURE EXTRACTION
First Input Frame from Video/ Camera
KEYPOINTS, RGB AND LBSP FEATURES FOR THE INITIALIZED BOUNDING BOX
ANCHOR POINT MODEL
WEIGHTED RGB HISTOGRAM
LBSP DESCRIPTOR
ADAPTATION OF CONSISTENCY OF A KEYPOINT BY EVALUATING ITS CLOSENESS
Addition and deletion of Keypoints
APPEARANCE MODEL UPDATER
SIMILARITY WITH RGB AND LBSP MODELS
Rest Input Frames from Video/ Camera
COMPARING KEYPOINT DESCRIPTORS
Coarse Target Prediction
VOTING BY KEYPOINTS IN SCORE MATRIX USING ITS ENCODED STRUCTURE AND CONSISTENCY
Object Location
OBSERVATION MODEL
OBJECT LOCALIZATION

Figure 5.2 Tracking using Numerous Anchor points (TUNA)

5.5 Tracking Using Numerous Anchor points

In this section our proposed tracker called TUNA (Tracking using Numerous Anchor points) is detailed. Figure 5.2 represents the block diagram of our tracking system. The main system components are *feature extractor*, *appearance model*, *observation model*, *object localization*, *consistency adaptation* and finally *appearance model updater*.

The term *anchor point* refers to a keypoint and vector pointing to the object center, along with its feature consistencies. Hence, it differs from a conventional keypoint feature, which does not have any relation with the object center. The tracking is executed as follows. In the first frame, keypoints are extracted and described for the initialized bounding box. These keypoints are modeled in a star-like structure with the object center as the root of the tree and their vector (Euclidean distances in X and Y) with respect to the center are encoded (See Figure 5.2). With this step, the construction of the *anchor point appearance model* is completed. At the same time, the global model is built by computing pixel-level LBSP and color RGB reference models. Then, keypoint features are detected and described in the next frame and are matched for similarity with the keypoints present in the anchor-point appearance model. This is the *observation model*, where the keypoint descriptors are matched for similarity using $L2$ norm. Then, each matched keypoint votes with its associated anchor point and its present location for the object center for the current frame. For *object localization*, all the individual votes are analyzed for maximum aggregation of votes, which represent the final object position. The *consistency adaptation* reflects the consistency of prediction of anchor points present in the model. The *long-term* consistency evolves over the tracking process and becomes larger if a keypoint is re-matched and predicts closer to the final target center and vice-versa. While the *short-term* consistency prevents abrupt change of object location predictions due to dynamic appearance changes. The *appearance model updater* computes for maximum similarity between the final tracking output obtained in previous step with the RGB and LBSP appearance models for deciding if the model should be updated or not. In this step, new anchor-points (keypoint features with their vector and consistency) are added to the anchor point model and poor keypoint features are removed from it based on their consistency.

5.5.1 Feature Extraction

In this step, *three* features are extracted : anchor-point features (SIFT [22] keypoints encoded with a vector pointing to the center of the object), color (RGB) and pixel level binary features (LBSP [60]). First, keypoints are detected and described for the bounding box and encoded into anchor points. SIFT keypoints are used as they are proven robust to illumination, rotation, scale etc. [22]. Any other keypoints can be used. Similarly, RGB histogram and LBSP descriptors are computed for the object contained in the bounding box. The LBSP is a 16-bit binary coded descriptor and provides pixel level modeling. For the RGB color model a weighted 3-D histogram for all the pixel values lying in the initialized bounding box is calculated. Hence, in the proposed tracking framework, three features are kept as reference models for the object to be tracked. For object localization only anchor-point features are used whereas, the color and pixel-level features are used during the anchor-point appearance model update.

5.5.2 Anchor Point Appearance Model

The filtered keypoints obtained from the initialized axis-aligned bounding box can be visualized in the form of a directed star graph-like structure denoted as $G(P, L)$, where vertices are directed towards the center. P represents the keypoints belonging to the object (See Figure 5.2) and edge L represents the connection between the vertices and the root of the structure. In our scenario, the vector of a keypoint is an edge, and is denoted as $L = [\Delta x_{k_i}]$, directed towards the center. Here, Δx_{k_i} contains the Euclidean distance of the keypoint's location x_{k_i} with respect to the center. Hence, the anchor point appearance model consists of the following :

- Descriptor of keypoint in the anchor point model
- Constraint Vector of a keypoint that describes its location with respect to the object center Consistencies denoted by L
- ST(short-term) and LT (long-term) Consistencies of a keypoint that indicates the keypoint's relevance for the object. A keypoint located nearby to the object center will have higher LT consistency as compared to others and is adapted online with a learning parameter during the tracking process. Further, the keypoint's ST will have a higher value if its individual prediction for the object center is closer to the globally voted object localization by all the keypoints present in anchor point model.

5.5.3 Observation Model

After the construction of the appearance model in the first frame, the keypoints are detected and described for the subsequent frame. Detecting keypoints all over the frame helps in finding an object having large or abrupt motion. Then, these keypoints are matched for similarity with the feature descriptors of the keypoints present in the anchor point appearance model by comparing their feature descriptors using $L2$ norm. For filtering bad keypoint matches, we use the ratio test of [22], and remove the matches that have a distance ratio of more than 0.9. Moreover, the mutual matching correspondence of keypoints between consecutive frames is confirmed, i.e, one-sided matched keypoints are not considered for voting and object localization. Only two-sided mutual matches are kept.

For the rest of the text in the paper, the matching of a keypoint will refer to matching of keypoint in the current frame with those keypoints present in the anchor-point appearance model.

5.5.4 Object Localization

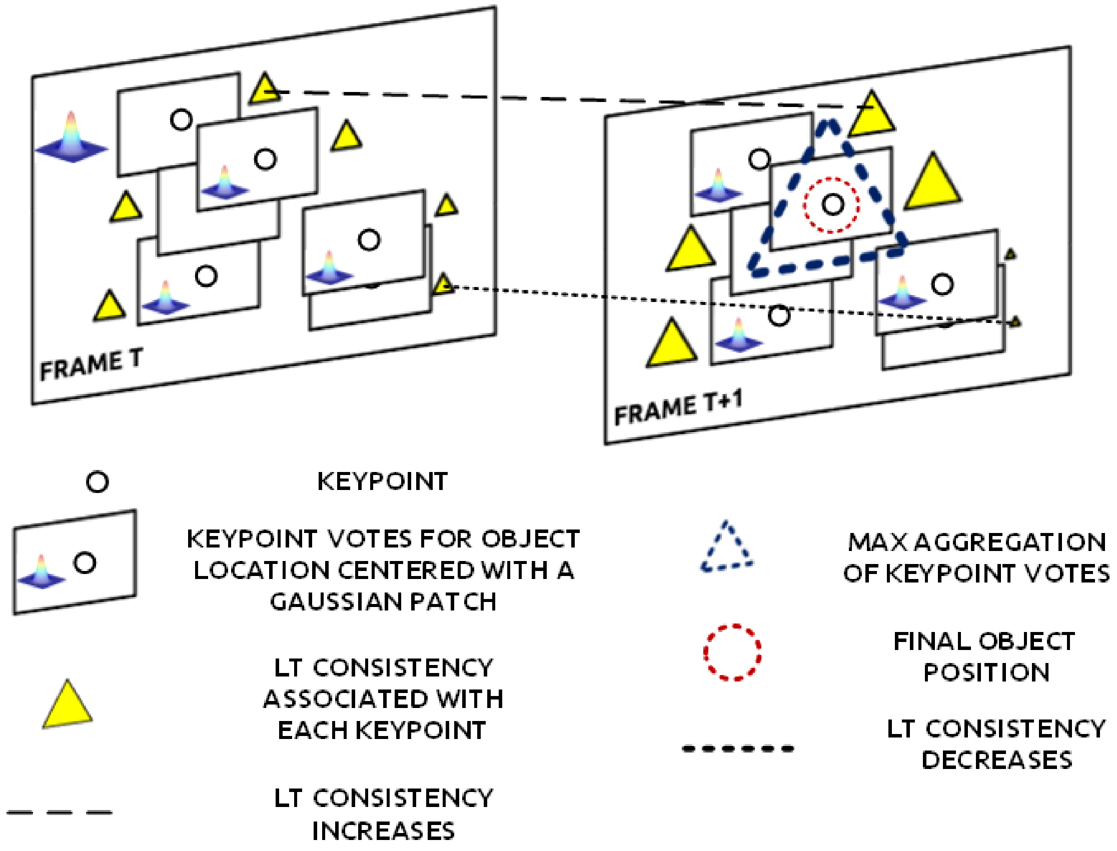

FRAME T
FRAME T+1
KEYPOINT
KEYPOINT VOTES FOR OBJECT LOCATION CENTERED WITH A GAUSSIAN PATCH
LT CONSISTENCY ASSOCIATED WITH EACH KEYPOINT
LT CONSISTENCY INCREASES
MAX AGGREGATION OF KEYPOINT VOTES
FINAL OBJECT POSITION
LT CONSISTENCY DECREASES

Figure 5.3 Visualization of keypoint voting and object localization. Here yellow triangles represent the consistency of a keypoint. Bigger yellow triangles represent higher consistency and vice-versa.

Consider visualizing the voting by a keypoint for the object center in the image space (See

Figure 5.3). The pixel location at which the keypoint is pointing for the object center, is centering a Gaussian patch which gives more value to the center than other pixel locations around it. The advantage of voting in a Gaussian patch helps to localize the object center even if the keypoint gets displaced from its original configuration in the anchor-point appearance model and thus is flexible towards deformation of object. Therefore, when a keypoint k_i is matched, it votes for the object center x, with its structured constrained vector (L_{k_i}) as :

$$P(x|k_i) \propto \frac{1}{\sqrt{2\pi|\Sigma|}} exp(-0.5(x - (L_{k_i} + x_{k_i}))^T \Sigma^{-1} (x - (L_{k_i} + x_{k_i})) \tag{5.1}$$

Here, $P(x|k_i)$ is the constraint vector score given by a keypoint, k_i for the object center, x, and Σ is covariance.

Each keypoint votes for the object center with its constrained vector score, its long-term consistency, and its short-term consistency as a total score in a Score Matrix, SM. Therefore, the total score for the object center can be formulated as a likelihood function, which is given by the dot product of the constrained vector score of a keypoint, its long-term consistency, and its short-term consistency. Hence, the likelihood expression as a function of total score by a keypoint can be written as :

$$SM(x) = \sum_{i=0}^{K} P(x|k_i).LT_{C_{k_i}}.ST_{C_{k_i}} I_{(k_i \in K)} \tag{5.2}$$

where, $LT_{C_{k_i}}$ is the long-term consistency, $ST_{C_{k_i}}$ is short-term consistency of a keypoint, and $I_{(k_{(i)} \in K)}$ is an indicator function, which is set for keypoints contained in the anchor-point appearance model that are matched in current frame. K is the total number of keypoints present in the anchor point appearance model. The cluster where the sum of individual scores is highest is taken as the final object center location, denoted as $x_{OCenter}$. The cluster shown a dashed blue colored triangle in Figure 5.3 represents that majority of keypoints are voting for the same object location. Hence, the final object location is given by :

$$x_{OCenter} = \arg\max\,(SM(x)|x \in SM) \tag{5.3}$$

5.5.5 Model Parameter Estimation

Long-term (LT) Consistency of a keypoint : It is estimated using a measure called *closeness*, $M_{C_{k_i}}$ associated with a keypoint, and is measured by computing the proximity of a keypoint's prediction for object center denoted as $x_{PredCenter_{k_i}}$, with respect to the final obtained object center using Equation 5.3. It is calculated using Equation 5.4 for the current frame T.

$$M^{t}_{C_{k_i}} = max((1 - |\alpha(x_{OCenter} - x_{PredCenter_{k_i}})|), 0.0) \tag{5.4}$$

Hence, a keypoint that predicts closer to the center will have higher closeness value, as compared to others. The keypoints which predicted very far from the final obtained center are assigned a value of 0.0, thus reducing their impact on voting for the object center for the future frames. This parameter is adapted for all the frames, as we will see in the next subsection. For the initial frame, closeness measure for all the keypoints present in the appearance model are initialized using Equation 5.5 as :

$$M^{t_0}_{C_{k_i}} = max((1 - |\alpha * L^{0}_{k_i}|), 0.5) \tag{5.5}$$

Here, $L^{0}_{k_i}$ is the initial vector associated with each keypoint k_i for frame T_0, and α is closeness factor. In the first frame LT consistency of a keypoint equals to $M^{t_0}_{C_{k_i}}$. The motive of using such an initialization function helps in assigning larger closeness value to those located keypoints that lie closer to the object center (indicating that the keypoints probably belong to the object) as compared to those which are farther (indicating that they may belong to the background).

Short-term (ST) Consistency of a keypoint : By analyzing how far away the keypoint predicted from the final object center obtained in frame t, the impact of k_i for future object center predictions in voting can be controlled. For instance, if a keypoint's prediction for the object center is very close to the object center obtained from Equation 5.3 in frame t, then its short-term consistency for frame $t + 1$ increases using Equation 5.6. But on the other hand, if a keypoint voted far from the object center, then its short-term consistency for prediction for object center reduces for frame $t + 1$. The advantage of analyzing short-term consistency aids in coping with sudden appearance changes of the object due to occlusion, rotation, illumination etc.. For instance, if a keypoint has a high long-term consistency and due to sudden appearance change, the keypoint votes incorrectly for the object center with a higher voting score. But if we analyze its short-term consistency for which its value will be less, therefore reducing its impact globally for the voting score in Equation 5.3 :

$$ST^{t+1}_{C_{k_i}} = exp\left(-\frac{(x^{t}_{PredCenter_{k_i}} - x^{t}_{OCenter})^2}{\eta}\right) \tag{5.6}$$

where η is a scaling factor.

5.5.6 Model Parameter Adaptation

In this step, the long-term consistency of a keypoint is adapted for all the keypoints that are present in the appearance model depending on their closeness measure. Keypoints that are matched more often, and for which their individual prediction is closer to the majority prediction obtained from Equation 5.3, will have larger closeness as compared to the rest of the keypoints that are predicting farther. This also provides an indication whether the keypoint belongs to the object or is a background keypoint, since if a keypoint does not predict for the center or if it is predicting very far, its *closeness* will be less and its long-term consistency will reduce eventually, according to Equation 5.7.

$$LT^{t+1}_{C_{k_i}} = \begin{cases} (1-\delta)LT^{t}_{C_{k_i}} + \delta M^{t}_{C_{k_i}}, & \text{if } I_{(k_i \in K)}\text{is true} \\ (1-\delta)LT^{t}_{C_{k_i}}, & \text{otherwise} \end{cases} \tag{5.7}$$

where, δ is an adaptation factor.

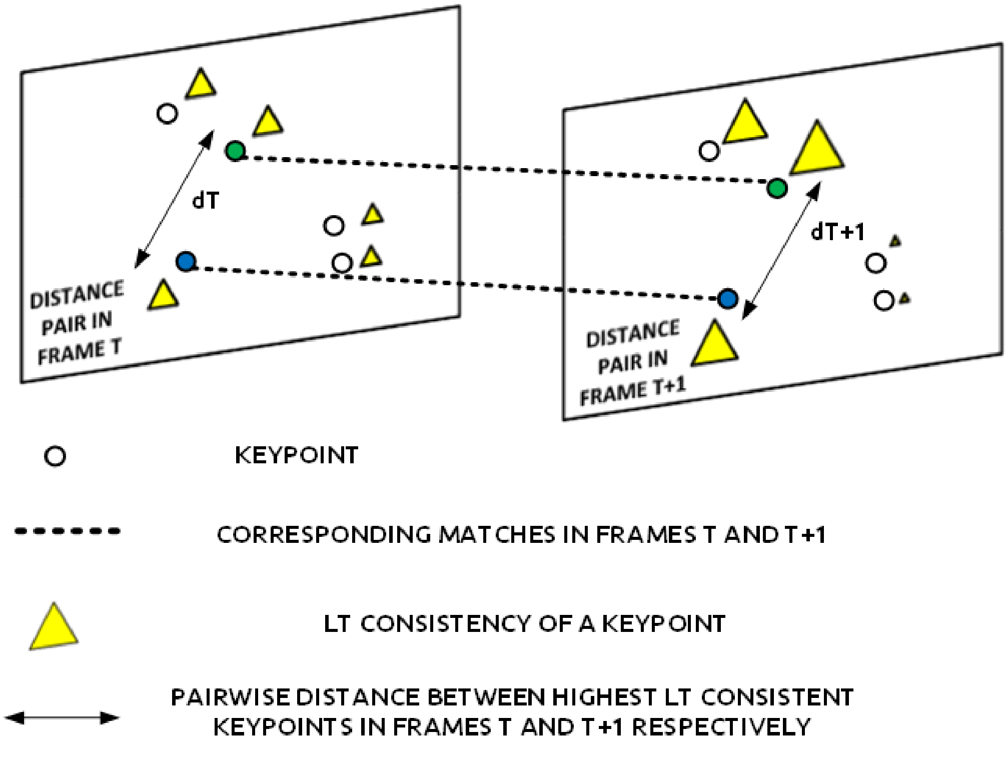

dT
DISTANCE PAIR IN FRAME T
dT+1
DISTANCE PAIR IN FRAME T+1
KEYPOINT
CORRESPONDING MATCHES IN FRAMES T AND T+1
LT CONSISTENCY OF A KEYPOINT
PAIRWISE DISTANCE BETWEEN HIGHEST LT CONSISTENT KEYPOINTS IN FRAMES T AND T+1 RESPECTIVELY

Figure 5.4 Scale Estimation

5.5.7 Appearance Model Update

Finally, the appearance model is updated only when a high tracking quality is achieved. The criteria for measuring the tracking quality is based on two features : the local pixel level LBSP (Local Binary Similarity Pattern) feature, and the global RGB color feature respectively. Only the anchor-point appearance model is used for object localization and is *updated* during tracking process based on matching similarity criteria of LBSP and RGB color features that are kept as *reference* models from the initial frame. Hence, after every object location by the tracker using the anchor-point model, the LBSP and RGB color models for the obtained bounding box are matched for similarity with their respective LBSP and weighted RGB reference models. The LBSP descriptor is matched for similarity using *Hamming* distance and weighted color histogram is matched for similarity using $L2$ norm respectively. The advantage of having a weighted color histogram, is to give more importance to the foreground pixels that are closer to the object center and less importance to the background pixels.

If the similarity comparisons agree with the reference models, then new anchor points are added to the anchor-point appearance model. The newly added keypoints are initialized with their respective structured constrained vectors and consistency values. The keypoints whose long-term consistency is poor and is lower than a threshold of $LT_{C_{min}}$, are removed from the model.

5.5.8 Scale Estimation

To adapt the scale to the current object location, we utilize a pairwise distance measure between keypoints that have been matched for similarity between two consecutive frames. This Euclidean paired distance represents the distance between keypoints and indicates how much the keypoint has moved due to scale change of the object. Moreover, by taking a mean of these paired distances, a single computed scale value can be applied to the bounding box. The number of keypoints that are considered for computing the pairwise distance depends on the total number of matches between two consecutive frames and their long-term consistency. The distance between the keypoint having the highest consistency (represented by blue color in Figure 5.4) with all other keypoints (represented by green color) are computed for frame T. Similarly, their corresponding distance is noted in frame $T + 1$.

Then, a distance ratio is computed for a keypoint pair and is given by $d(T + 1)/dT$ and a mean value is computed. The final scale change is applied to the bounding box after a period of every ten frames. Moreover, it is only applied when the mean lies within ± 10 % of the initial size of the target object. This is because we assume that the scale of the object would not undergo such an abrupt difference in scale between two consecutive frames. Note, the scale estimation is not limited to a fixed aspect ratio of the object.

5.6 Additional details on the working of TUNA

Due to partial occlusion, some keypoints become latent (not visible) during the tracking process. Therefore, only the keypoints having indicator function, $I_{(k_{(i)} \in K)}$, as one can be tracked. These keypoints act independently for object prediction and vote for the object center with their vector and their consistencies. Together the LT and ST consistencies associated with features helps in voting for object localization, since the consistent performing features only vote in the score matrix with their associated consistencies.

For some frames, even if there are no matches due to a long-term occlusion, motion blur or an out-of-plane rotation, the last obtained object location is not updated until the object appears again and the consistent keypoints present in the anchor-point model starts predicting. Refraining from updating the location during this time, helps in making less location errors. Together the LT and ST consistencies prevent abrupt prediction changes when the object undergoes large appearance variations during tracking. For example, it may be possible that a background keypoint having LT consistency is present in the anchor-point model, and is predicting wrongly for the object center. But, while evaluating its ST consistency, its value is lower for the next frame, since it predicted farther from the object center obtained using Equation 5.2. Hence, when it votes again for object center in the next frame with its consistencies, the voting score reduces in the score matrix for the next frame. This is because the LT consistency reduces due to its adaptation by learning factor, according to Equation 5.7 and the ST consistency reduces, according to Equation 5.6 respectively. This helps in preventing erroneous object location predictions.

Further, during object deformation some keypoints may get displaced, therefore when a keypoint votes for the object centering a Gaussian patch, the gaussian acts a flexible window for the keypoint displacement. Hence, the higher value assignment to the center as compared to rest of the pixels surrounding the keypoint makes TUNA tolerable to deformations. Further, the anchor-point appearance model with constrained vector is distinctive and helps to deal with distractors and background because the model captures the pattern of the local information of the object using keypoint descriptor and the global information of the object with the keypoint constrained vector.

5.7 Evaluation

The tracker performance is evaluated on a recent benchmark [68] having 51 video sequences. The video sequences have several attributes like severe illumination changes, abrupt motion changes, object deformations and appearance changes, scale variations, camera motion, long-term scenarios and occlusions. Our results are compared against other classic tracking algorithms : Multiple Instance Learning (MIL) [5], Color-based Probabilistic tracking (CPF) [87], Circulant Structure of tracking-by-detection with Kernels (CSK) [66], Kernel-based object tracking (KMS) [88], Semi-supervised on-line boosting for robust Tracking (SemiT) [84], real-time Compressive Tracking (CT) [82], Beyond Semi-Supervised Tracking (BSBT) [89], Robust Fragments-based Tracking using the integral histogram (Frag) [61], Tracking-Learning-Detection (TLD) [42], Mean-Shift blob tracking through Scale space (SMS) [90], Online Robust Image Alignment via iterative convex optimization (ORIA) [91], visual tracking via Adaptive Structural Local sparse Appearance model (ASLA) [71], and Incremental learning for robust Visual Tracking (IVT) [62], respectively.

5.7.1 Quantitative Evaluation

The evaluation is done using the standard evaluation protocol suggested by [68], which uses two criteria. The first is precision, where position error between the center of the tracking result and that of the ground truth is used. A threshold of 20 pixels is used for ranking the trackers. This threshold represents the percentage of frames for which the tracker was less than 20 pixels from the ground truth. The second is success that represents the bounding box overlap of the tracking result with the ground truth. The overlap is the ratio of intersection and union of predicted bounding box with the ground truth bounding box. Instead of using the standard threshold of 0.5, this benchmark uses AUC (Area Under Curve) and the threshold is varied from 0 to 1 and the AUC across all the thresholds is reported as success results. A larger AUC indicates higher accuracy of the tracker.

We tested three versions of our proposed tracker TUNA viz., first using anchor point model for object localization and LBSP features as reference model for appearance update, and the second using anchor point model for object localization and RGB feature as reference model for appearance and third one using anchor point model for object localization and LBSP and RGB features, both as reference models for appearance update. We remark that by using the third version, the overall precision of the tracker increases (See Table 5.1 and Figure 5.5). In addition, when tested without scale estimation version, the precision and success of the tracker reduces a little. TUNA performs second after TLD which emphasizes

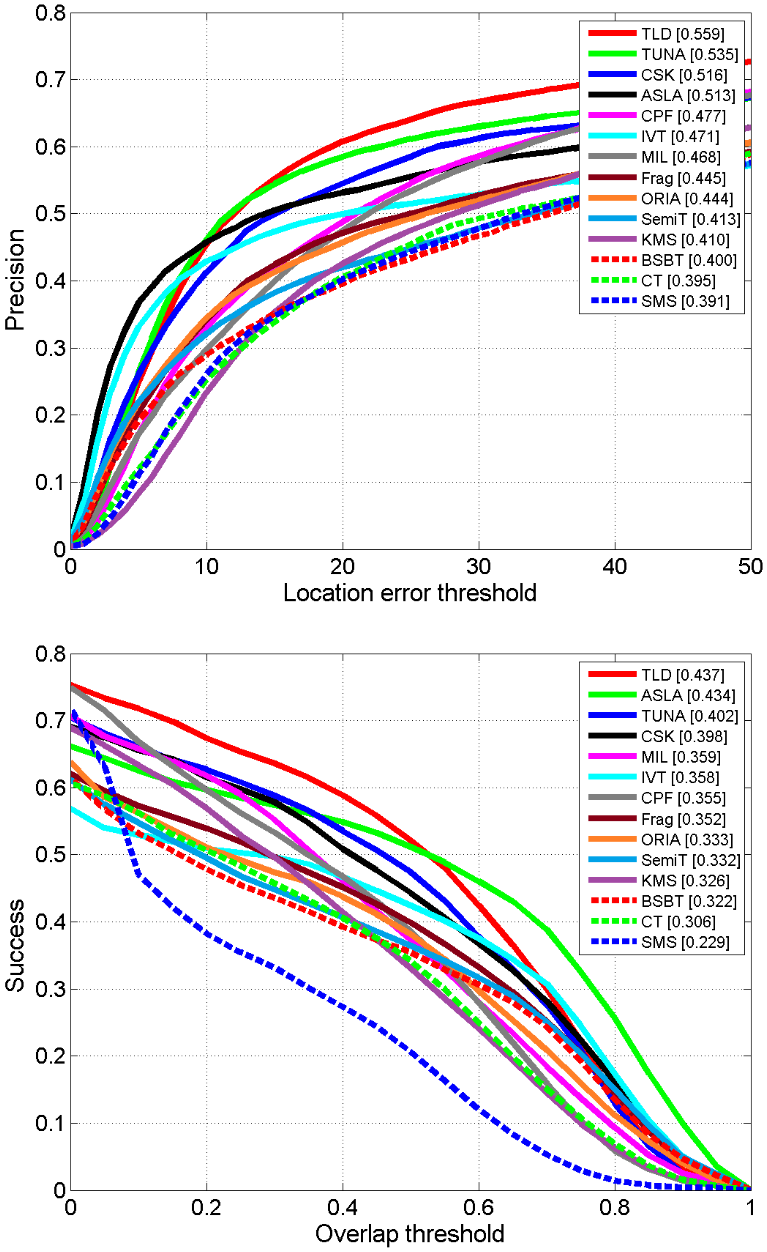


Figure 5.5 Precision and Success plot on all 51 video sequences. The proposed tracker TUNA outperforms several other state-of-the-art trackers. Best viewed in color and zoomed in.

Table 5.1 Summary of Experimental Results on 51 video dataset. The bold italic represents the best results and bold represents the second best results.

Algorithm	*Mean Precision 20 px*	*AUC*
TUNA (Proposed)		
Anchor Point Model + LBSP	53.0%	**40.9**%
Anchor Point Model + RGB	51.7%	38.1%
Anchor Point Model + LBSP + RGB	**53.5**%	40.2%
TUNA (Without Scale)	52.4%	39.9%
CSK [66]	51.6%	39.8%
MIL [5]	46.8%	35.9%
TLD [42]	***55.9***%	***43.7***%
Frag [61]	44.5%	35.2%

that detection module is an important engineering component in the tracker. TUNA[1] is implemented in C++ using OpenCV 3.0.0 library[2]. It runs at mean FPS of 8 (computed over the 51 sequences) on Intel Core i7 @ 3.40 GHz, 8GB RAM computer. The parameters used in all experiments for TUNA are summarized in Table 5.2.

Table 5.2 Parameters used in all Experiments

TUNA Parameters	Value
Closeness Parameter	$\alpha = 0.005$
ST Consistency Parameter	$\eta = 5000.0$
LT Consistency Initialization	$\lambda = 0.5$
LT Consistency Adaptation	$\delta = 0.1$
LT Consistency Min. Threshold	$LC_{min} = 0.1$

5.7.2 Attribute Wise Analysis

As seen from Table 5.3, low resolution (LR) severely affects the performance of most of the trackers. But our proposed tracker TUNA performs the best among all, showing the superiority of the anchor point model. Even in videos with low resolution, keypoint features can be extracted and thus encoding the structure of the object. Hence, keypoints with the structure votes accurately for the object location. Unlike other trackers that perform poorly, CSK that uses densely sampled features, can cope up.

[1] https ://bitbucket.org/tanushri/tuna

[2] http ://opencv.org/

Among all the other attributes, TLD performs better than other trackers showing its importance on its re-detection and failure module engineered in the tracker. Nevertheless, such component can further improve the performance of our proposed tracker, but TUNA still proves its superiority over TLD in low resolution (LR) and performs competitively on other challenging sequences performing as second best.

TUNA is able to perform very well in video sequences having motion blur (MB) due to the following facts. As each keypoint votes for the object location is associated with a LT and ST consistency, which is adapted during the tracking process, it helps to avoid too many wrong predictions. For instance, if a keypoint is LT consistent but if its ST consistency is too low, its voting contribution in score matrix for object location reduces. This also indicates that a keypoint from background (or an outlier) might be predicting for the object center wrongly, if it is included in the model. Thus, it is better to have few good predictions rather than having too many false predictions for object location. Moreover, maintaining a holistic color model and local pixel level helps in preventing unwanted model update, therefore preventing the model from drifts. TUNA also performs well on videos having fast motion (FM) as keypoints are detected all over the frame. Therefore, matching for object location is performed on a larger search region, unlike ASLA where the search region is limited due to particle filter.

For videos with occlusion (OCC), TLD performs best due to its re-detection scheme. Note that the color features prove their distinctiveness for occlusion with CPF tracker. TUNA performs competitively here ranking as third among others. This is because even if some keypoints become hidden due to occlusion, the independent acting keypoints in the anchor point model votes for the object center with their consistencies. Moreover, the keypoints from the background will have smaller LT consistency and smaller voting contribution as compared to the foreground keypoints. Hence, there are fewer chances for incorrect object prediction during partial occlusion.

The proposed tracker is able to handle object deformation (DEF) very well. This is because when a keypoint is matched, it votes with the anchor points (that has the constrained vector structure of a keypoint) centered with Gaussian patch. Hence, even if the keypoints gets displaced due to object deformation, the gaussian patch allows voting in a neighborhood with more emphasis on the center pixel, which makes it handle the error associated with the keypoint deformation. Hence, for some frames, even if there are no matches due to long-term occlusion and out-of-plane rotation, the last obtained object location is not updated until the object appears again and the keypoints start predicting, thus making erroneous location errors.

The pairwise ratio distance between keypoints helps to gauge the scale change between two frames accurately by analyzing the LT consistency of keypoints. Moreover, the scaling technique does not take into account any fixed aspect ratio and thus can be applied to objects of various sizes. TUNA ranks third among the state-of-the-art trackers for scale variations (SV). The videos with background clutter (BC) also impacts the performance of all trackers except CSK, showing dense sampling of negative features around the object helps to better discriminate the object from background.

5.7.3 Qualitative Evaluation

To better demonstrate the performance of TUNA, snapshots for some challenging video sequences are presented in Figure 5.6. Note that TUNA tracks successfully object in long video sequences like *doll* and *lemming* that contain more than 1000 frames. This is because of the property of anchor point model that remembers the holistic appearance of the object. Moreover, the keypoints that are matched frequently with higher LT and ST consistency, helps to track the object till the end of the sequence. Moreover, the parameter adaptation of LT consistency associated with keypoints in the model, helps to retain relevant features and remove unreliable features from the model.

5.8 Conclusion

In this paper, an online adaptive model-free tracker with a novel anchor point appearance model is proposed. The keypoints are assembled into anchor-point features that are arranged in a star graph-like structure with the object center. All the anchor points in the structure votes for the object center and the object localization is done by analyzing the maximum of these voting scores by every keypoint in a score matrix. Our results prove that the anchor point model with constrained structure acts as a robust feature for visual object tracking, specifically for tracking objects in low resolution, motion blur, or having deformation or abrupt motion. The voting by a keypoint with a Gaussian helps to tackle the deformation of the object. The dynamic adaptation long-term consistency and short-term consistency of a keypoint helps in stable and accurate object localization. For the adaptation of scale, a new keypoint pairwise distance measure is proposed. It does not involve complex geometrical or rotation calculation, unlike existing methods. Finally, the crucial update of the system is governed by finding similarity of the local pixel level binary features and global weighted color features reference models. Along with this, the features are added and removed from the anchor-point appearance model based on their LT consistency. Nevertheless, the robustness of the proposed tracking approach relies on the keypoint detection step. An interesting direction for future work is to extend the proposed tracker with a detection framework, which may improve further the performance of the tracker.

Acknowledgment This work was supported in part by FRQ-NT team grant #167442 and by REPARTI (Regroupement pour l'étude des environnements partagés intelligents répartis) FRQ-NT strategic cluster.

Table 5.3 Comparison with state-of-the-art trackers on videos having attributes : Motion Blur (MB), Fast Motion (FM), Background Clutter (BC), Deformation (DEF), Illumination Variation (IV), In-plane Rotation (IPR), Low Resolution (LR), Occlusion (OCC), Out-of-plane-Rotation (OPR), Out-of-View (OV), Scale Variation (SV). The bold italic represents the best results and bold represents the second best results.

					Mean	Precision					
Video Att.	MB	FM	BC	DEF	IV	IPR	LR	OCC	OPR	OV	SV
TUNA**(Prop.)**	**0.476**	**0.452**	0.348	**0.487**	0.415	0.463	***0.438***	0.486	0.487	**0.474**	0.512
MIL [5]	0.338	0.382	**0.450**	0.447	0.387	0.448	0.168	0.427	0.461	0.390	0.462
CPF [87]	0.298	0.365	0.402	***0.488***	0.386	0.456	0.134	**0.501**	**0.510**	0.455	0.464
CSK [66]	0.346	0.362	***0.534***	0.440	0.469	0.513	*0.437*	0.475	0.506	0.361	0.494
KMS [88]	0.372	0.359	0.391	0.404	0.384	0.375	0.232	0.401	0.401	0.385	0.408
SemiT [84]	0.339	0.352	0.368	0.421	0.309	0.371	0.432	0.391	0.383	0.314	0.376
CT [82]	0.316	0.333	0.327	0.418	0.352	0.368	0.138	0.406	0.390	0.348	0.419
BSBT [89]	0.330	0.329	0.329	0.372	0.324	0.388	0.244	0.393	0.400	0.415	0.345
Frag [61]	0.274	0.323	0.404	0.444	0.320	0.388	0.147	0.441	0.426	0.324	0.379
SMS [90]	0.299	0.321	0.327	0.417	0.346	0.332	0.170	0.402	0.401	0.337	0.400
TLD [5]	***0.482***	***0.517***	0.420	0.469	**0.497**	***0.545***	0.339	***0.518***	***0.546***	***0.553***	***0.562***
ORIA [91]	0.246	0.276	0.377	0.342	0.408	0.479	0.236	0.431	0.466	0.323	0.431
ASLA [71]	0.283	0.270	0.484	0.426	***0.499***	**0.501**	0.174	0.444	0.500	0.322	***0.539***
IVT [62]	0.220	0.219	0.395	0.389	0.387	0.435	0.272	0.430	0.435	0.290	0.473

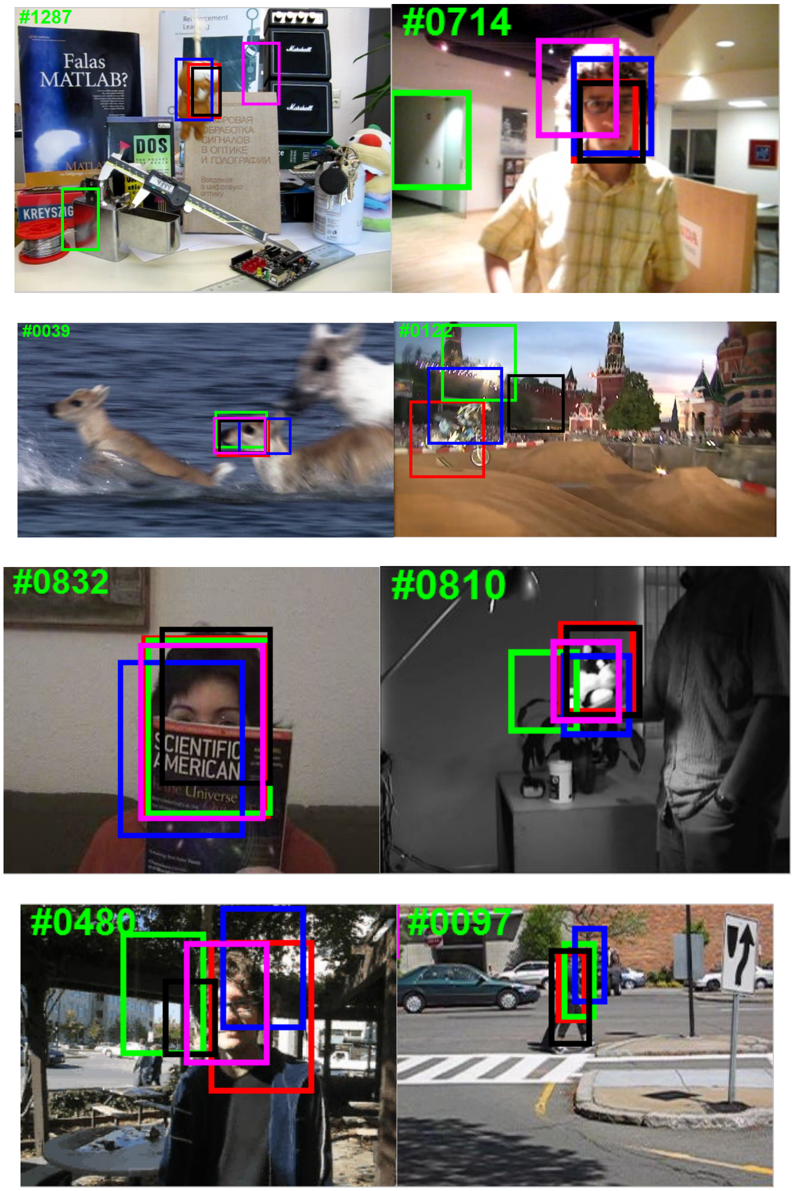


Figure 5.6 Snapshots results of selected tracking algorithms on video sequences : *lemming*, *david*, *deer*, *motorcycling*, *faceocc1*, *sylvester*, *trellis* and *couple* respectively.

🟥 TUNA 🟪 CSK 🟦 MIL ⬛ TLD 🟩 Frag

CHAPTER 6 ARTICLE 3 : ROBUST FACE TRACKING USING MULTIPLE APPEARANCE MODELS AND GRAPH RELATIONAL LEARNING

Authors

Tanushri Chakravorty, Guillaume-Alexandre Bilodeau,
LITIV Lab., Polytechnique Montréal
Eric Granger, *LIVIA, École de technologie supérieure, Montréal*

E-mail : {tanushri.chakravorty, gabilodeau}@polymtl.ca, eric.granger@etsmtl.ca

Submitted to Image and Vision Computing, 29 June, 2017

Reference as T. Chakravorty, G. A. Bilodeau and E. Granger, "Robust Face Tracking using Multiple Appearance Models and Graph Relational Learning," *arXiv preprint arXiv :1706.09806*

Abstract

This paper addresses the problem of appearance matching across different challenges while doing visual face tracking in real-world scenarios. In this paper, FaceTrack is proposed that utilizes multiple appearance models with its long-term and short-term appearance memory for efficient face tracking. It demonstrates robustness to deformation, in-plane and out-of-plane rotation, scale, distractors and background clutter. It capitalizes on the advantages of the tracking-by-detection, by using a face detector that tackles drastic scale appearance change of a face. The detector also helps to reinitialize FaceTrack during drift. A weighted *score-level fusion* strategy is proposed to obtain the face tracking output having the highest fusion score by generating candidates around possible face locations. The tracker showcases impressive performance when initiated automatically by outperforming many state-of-the-art trackers, except Struck by a very minute margin : 0.001 in precision and 0.017 in success respectively.

Keywords : face tracking, multiple appearance models, L2-subspace, graph relational learning, weighted fusion

6.1 Introduction

Face tracking has been studied for decades and it is still one of the challenging problems in computer vision. Face tracking in unconstrained videos promises to augment a wide range of applications in robotic vision, video analysis and face recognition, and is not only limited to visual surveillance. It is often used in video conferencing, but it is also useful in video-based face recognition as shown in [14]. It is defined as the task of locating a face in a given frame whether it is occluded or not. The face tracker is initiated in two ways : (1) using a ground-truth bounding box containing a face, (2) using a bounding box provided by a face detector. This box is also called an ROI (Region Of Interest). The output of the face tracker is the location of a face in a frame and is represented by a bounding box.

As the face tracker outputs ROIs over a series of consecutive frames in a video sequence, it accumulates multiple evidence for the presence of a target face. Hence, the face tracker can preserve the identity of a target face since it works on the principle of *spatio-temporal* information between consecutive frames. In contrast, a face detector searches for a face in the entire image, without any *spatio-temporal* information, and thus cannot keep the identity of a face.

Our primary contribution is to represent a face in a L2-subspace with a *relational* graph. The term *relational* describes the relation of features with the center of the bounding box during tracking initialization. This information comprises of three components : L2 distance of a feature with the center (FDL), importance of the feature (w), and feature descriptor (D). This model not only describes the appearance of the target face by representing it in a L2-subspace, but also encapsulates semantic information specific to the target face for occlusion. Thus, when this relational graph is discovered in a subsequent frame by matching feature descriptors, each matched feature outputs a center location of the target face using its L2-subspace representation. This center prediction is approximated by using multiple kernels in a response map reflecting the importance of each matched feature for the center prediction. The face localization is done by first concatenating all the generated kernel responses and then analyzing the peak response in the kernel map, which is transformed back to the cartesian coordinate system as face center location. Analyzing the peak in the map helps in eradicating the influence of errors during face localization, since multiple overlapped responses indicate reliable face center prediction over responses generated by tracking errors.

The relational graph is learned incrementally by adding and deleting connections in the graph during the *appearance* model update. Since the good connections are retained in the graph to help in localizing the center of the target face, this appearance model acts like a long-term memory of the target face. This appearance model is coined as GRM (Graph Relational Model), and is one of the proposed appearance models used in FaceTrack. The graph matching and face localization concept using GRM is illustrated in Figure 6.1.

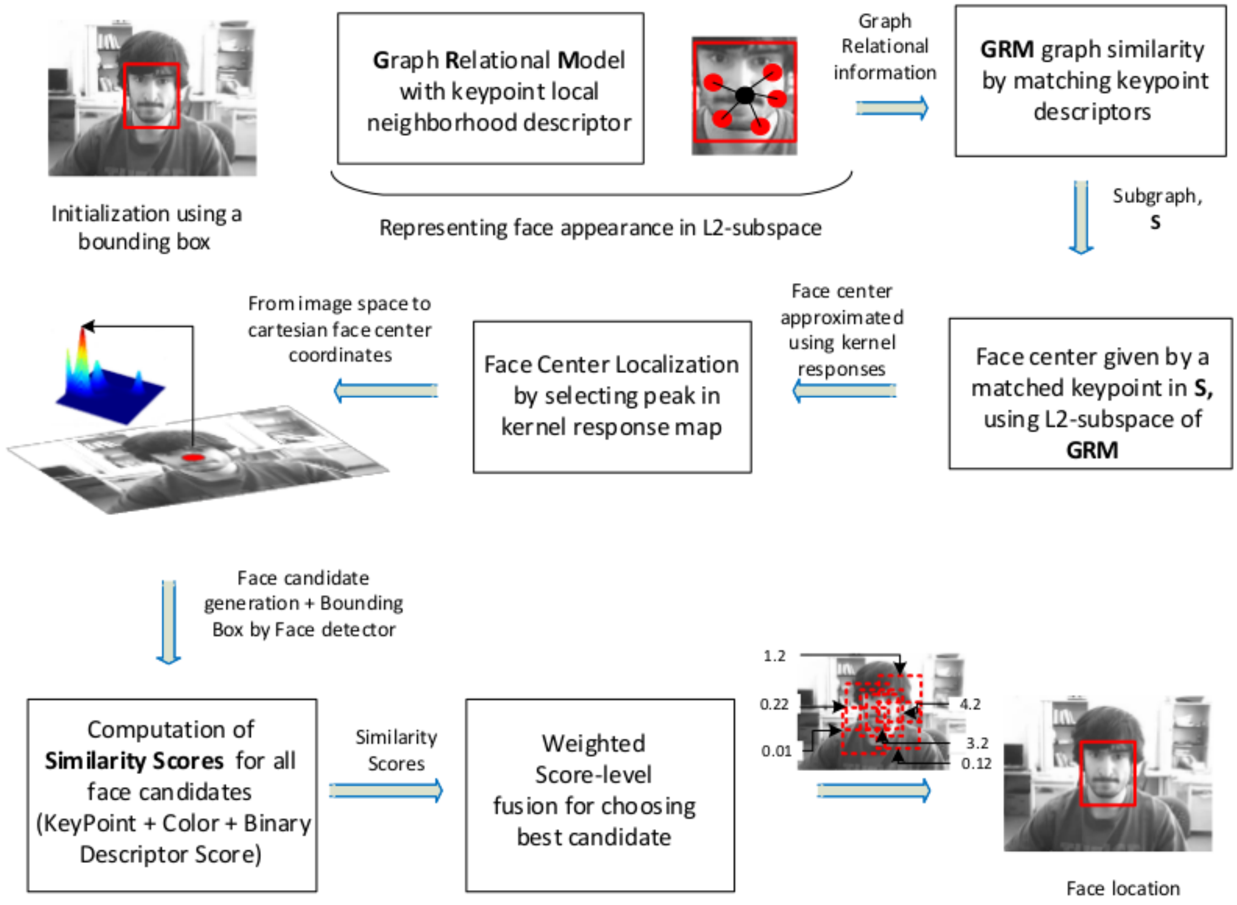

Initialization using a bounding box
Graph Relational Model with keypoint local neighborhood descriptor
Representing face appearance in L2-subspace
Graph Relational information
GRM graph similarity by matching keypoint descriptors
Subgraph, S
Face center given by a matched keypoint in S, using L2-subspace of GRM
Face center approximated using kernel responses
Face Center Localization by selecting peak in kernel response map
From image space to cartesian face center coordinates
Face candidate generation + Bounding Box by Face detector
Computation of Similarity Scores for all face candidates (KeyPoint + Color + Binary Descriptor Score)
Similarity Scores
Weighted Score-level fusion for choosing best candidate
1.2
0.22
0.01
4.2
3.2
0.12
Face location

Figure 6.1 Face localization process using Graph Relational Model.

In contrast, the other proposed appearance models, ICM (Isotropic Color Model) and BDM (Binary Descriptor Model), help to find the target face during drastic appearance changes like illumination variation, in-plane rotation, out-of-plane rotation and heavy occlusion. The ICM describes the holistic face appearance, whereas the BDM helps to detect the intrinsic spatio-temporal changes happening at the pixel level. They both serve as a short-term memory of the current target face appearance, and are updated *partially (and/or fully)*, depending on the occlusion detection strategy. By following this appearance model scheme for tracking, the *temporal* information of a target face gets accumulated, and the tracker gets an appropriate appearance memory of the target face for appearance matching.

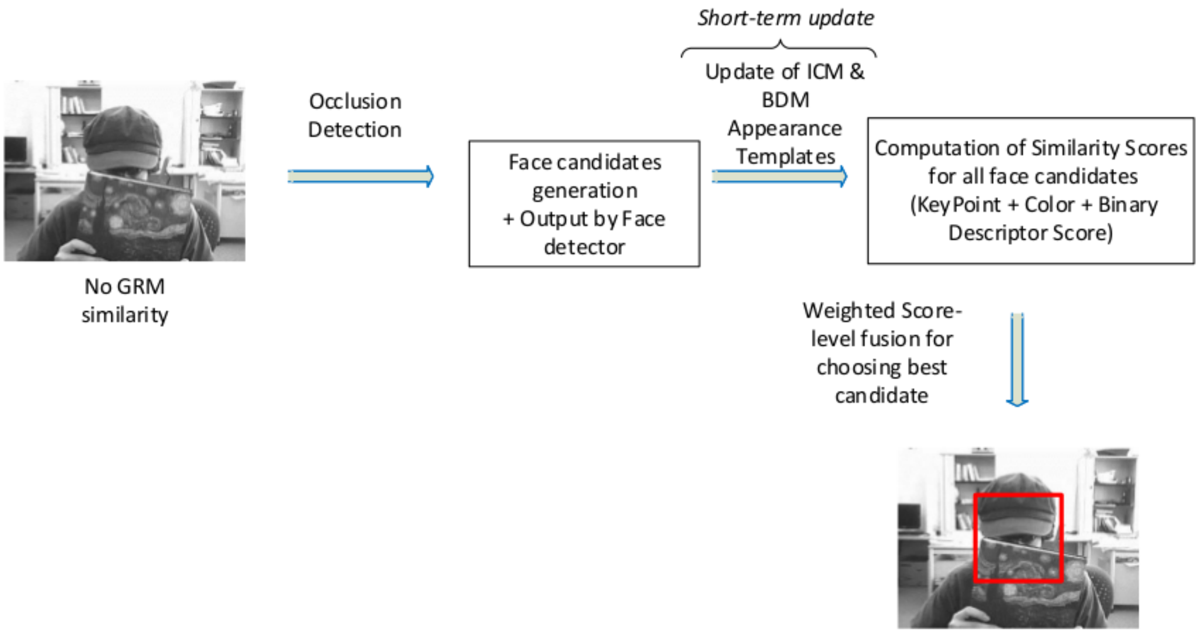

Short-term update
Update of ICM & BDM Appearance Templates
Occlusion Detection
Face candidates generation + Output by Face detector
Computation of Similarity Scores for all face candidates (KeyPoint + Color + Binary Descriptor Score)
No GRM similarity
Weighted Score-level fusion for choosing best candidate

Figure 6.2 Occlusion detection, tracking control and update strategy for the proposed face tracking system using GRM.

The GRM is effective as long as the graph structure remains visible and gets fully or partially matched. During other situations, the remaining appearance models (ICM & BDM) are used for estimating the face location as shown in Figure 6.2. Apart from this, the appearance models are incrementally learned and the importance of features is determined on-the-fly for keeping a temporal memory, both long-term (GRM) and short-term (ICM and BDM), of the appearance of the target face. The proposed model is built to handle many tracking challenges like motion blur, fast motion, partial and heavy occlusion, background clutter and scale change. Each component plays a vital role in localizing the target face and the proposed tracker utilizes all the advantages from these components for accurate tracking.

Our secondary contributions are a robust tracking strategy that assigns importance to appearance features during tracking *initialization* and continues during the *entire* face tracking process. The robustness is integrated using *isotropy* to the appearance features used in tracking. The isotropic nature of features is formulated in a manner such that the feature closest to the center obtains the highest *importance* as compared to others. By doing this, the background features that may get encapsulated in the appearance model, have lesser contribution in the kernel response map for target face center localization. In addition, the importance of the features get *adapted* online and the lesser important features are deleted from the graph and the newer ones are added during model update, following the same policy of using isotropy to establish the importance to newly added features.

Apart from this, we use a *tracking-by-detection* approach by employing a face detector, [16], with FaceTrack. The face detector helps to handle scale and aspect ratio changes of the face, drift and may help in reinitialization of the tracker during severe appearance changes. But, using either a single or multiple appearance based tracker with a face detector alone cannot effectively solve the face tracking problem. This is because the face detector focuses only on appearance similarities and ignores the spatio-temporal information in images, due to which there are large fluctuations in detection scores between two consecutive frames. On the other hand, the tracker might lose the target face due to large appearance variations. Hence, the face detector output is also used in face localization, thus capitalizing on their respective strengths.

However, due to tracking noise and face deformation the localized face may not be precise. Hence, face candidates are generated around the localized face region obtained using face appearance matching with the help of multiple appearance models. Thus, in the proposed method, face tracking is considered as a problem of accurately estimating the *face candidate* having the highest fusion score in a given frame. Hence, to obtain the final tracking output, a weighted *score-level fusion* criteria is formulated for selecting the best face candidate.

6.1.1 Contributions

The main contributions of this paper are as follows :

1. A novel face tracking method is proposed that utilizes multiple appearance models to account for the temporal appearance matching of a target face for robust tracking.
2. A long-term and short-term strategy is proposed for effective matching during face tracking in real-world unconstrained video sequences.
3. Robustness to face appearance features is integrated using isotropic weights. This ensures to obtain face localization using importance face appearance features during the entire tracking process, thus tackling drift and background clutter.
4. A weighted score-level fusion approach is proposed for estimating the best face candidate as face location.
5. A novel tracking control and update strategy that accounts for occlusion detection, tracking robustness and stability is proposed.

The rest of the paper is organized as follows : Section 6.2 presents some related works in visual object tracking. Section 6.3 discusses the proposed tracking framework in detail. Section 6.4

provides the details of quantitative and qualitative experiments and analysis of each of the tracking method. Finally, Section 6.5 concludes this paper.

6.2 Related Work

In this section, we focus on the visual object tracking works related to the class of discriminative appearance-based trackers. These discriminative appearance-based trackers behave like binary classifiers and distinguish the target object from the background. These discriminative trackers incorporate some form of model update during the visual tracking process and the classifier learns from samples online [[42], [73], [79], [5]].

The TLD [42] method uses a binary feature detector and an optical flow tracker. The detector learns from the examples which are sampled online from the bounding box. Positive examples are labeled from the region inside the box and the negative examples are taken from the region around the bounding box. In contrast, MIL [5] utilizes Haar features as samples which are grouped into a bag. Along with the bounding box, the tracker uses rectangular windows around the nearby region as positive samples, since the target region can include some background region. Negative bags comprise of rectangular boxes which are farther from the bounding box. In Boosting [73], the method employs a boosting classifier based on Haar features for selecting discriminative features for distinguishing the target object from the background. In Struck [79], Haar features from the box are considered as an appearance model for tracking. In their method, instead of generating samples from around the bounding box, the samples are generated by translating the bounding box and then fed to a SVM (Support Vector Machine) classifier. Thus, the sampling strategy for Struck is different from the aforementioned tracking methods. However, the classifier learning is constrained by maintaining a budget that helps to maintain a set of the support vectors. Recently, correlation filter learning method like [66], has shown impressive results due to its dense feature extraction and sampling technique for high-speed tracking.

Detections provided by object detector are used in tracking objects whose prior information is known. The trackers of [42] and [73] are special cases of tracking-by-detection. The detections enable the tracking process to tackle scale appearance change and sometimes drift. Similarly, a face detector is used in FaceTrack to tackle drastic appearance change such as scale change between two consecutive frames and reinitializes it during drift.

Ross et al. introduced incremental subspace learning in visual object tracking with the concept that the target can be represented in a low dimensional subspace that can be helpful in dealing with tracking nuisances, like pose and illumination variation [62]. This idea works well in situations where the errors are small and localized, i.e., they follow a Gaussian distribution. However, in some scenarios like when there is occlusion, the errors might be large. In such cases, this type of global representation might not be able to cope up and thus result in track loss. To overcome this, the authors in [92], assumed that tracking errors follow a Gaussian-Laplacian distribution. Owing to their success, their error-removing method is employed in various works [93], [5]. In real-world scenario the data can however, contain various types of noise, and the data or noisy samples that may belong to other targets may get included in the appearance model of the target and ultimately degrade the performance, particularly for graph-based learning methods. Hence, authors in [94] proposed a spectral clustering method, which consider edges with higher weights in the graph cluster and segment other parts in the graph.

In our approach, the GRM model adds new samples to the relational graph by taking keypoints from the tracking bounding box itself, thus removing the need for using segmentation and clustering.

Besides this, adaptive appearance models like [39], [95], use face tracking for face recognition purpose. They use online samples for updating the appearance of the face, and employ forgetting factor for adapting the appearance model. Related to our work are object trackers [2], [1] and [80] that utilize structure of the object as the appearance representation for tracking.

In contrast to these approaches, our method maintains a temporal appearance memory using multiple appearance models of the target face that leverages the benefit of both long-term and short-term appearance updates that are proven essential for robust face tracking. Moreover, we adapt the face appearance representation such that potentially distracting regions are suppressed in advance and thus, no explicit tracking of distractors (similar looking faces) is required, thus ensuring stable face tracking in real-world scenarios. The next section details the workings of FaceTrack.

6.3 Proposed Face Tracking Method

It has been shown in [1] and [80] that structure can be a powerful appearance representation for visual object tracking, whereas in [2], it has been shown that the structure of an object can help to tackle occlusion. Our motivation for using structure is inspired by the idea that by exploring the *intrinsic* structure of a target face may help to discover a particular pattern of a face of interest.

In machine learning tasks such as subspace learning, semi-supervised learning and data clustering, informative directed or undirected graphs are used to study the pairwise relationships between data samples that helps to identify a pattern belonging to a specific object [37]. Thus, for identifying a particular pattern belonging to a face of interest, a graph has the following characteristics that can be highly beneficial for face tracking :

— Distinct Representation : Graphs are powerful representation tools. Higher dimensional data can be represented in a manner which can be utilized for problem solving.

— Relational information : Graphs can help to identify the internal structure which can be utilized by relational information such as metric between points in the graph, rather than just the attributes of the entities being present [96].

— Sparsity : Findings in subspace learning [97] show that sparse graph characterizes local relations and thus can help in better classification.

Hence, the aforementioned advantages of a graph can be used for building a robust appearance model for face tracking in videos.

Table 6.1 Multiple Appearance Models used in FaceTrack

Appearance Model	*Notation*	Feature Description
Graph Relational Model, **GRM**	D	SIFT keypoint features at each location and scale in an image, thus are multi-scale and spatially specific with their invariant keypoint descriptor [98].
	FDL	Represents face appearance in L2-subspace by encoding *L2 distance* of a SIFT keypoint from the tracked bounding box center, denoted as $FDL = [\Delta x, \Delta y]$.
	w	Describes the importance of a keypoint assigned using *isotropy*.
Isotropic Color Model, **ICM**		Holistic discriminative feature of the face (tracked bounding box), 3-channel *Gaussian* weighted color histogram (pixels are assigned importance using *isotropy*.)
Binary Descriptor Model, **BDM**		Encodes spatio-temporal local neighborhood information of a pixel into a 1-channel, 16-bit LBSP binary descriptor, [77].

6.3.1 Building the multiple appearance models

The proposed model characterizes the target face contained in the initialized bounding box by using multiple appearance models namely GRM, ICM and BDM respectively. GRM characterizes the face from two perspectives. First, by encoding features related to a face by detecting and describing keypoint descriptors that belong to the face. Second, by representing the keypoints in the L2 subspace by forming a relation between the detected keypoints with the center of the initialized box using relational information. It is robust towards partial occlusions and deformations, as a visible part of the GRM can still output the target face center by using its relational information. For more robustness, during the model initialization, the keypoints that are closer to the center are given higher *importance* and are assigned higher *weights* as compared to others that are farther from the center. The weight associated to the importance is given by Equation 6.1 :

$$w_{k_i} = \max((1 - |\eta \cdot FDL|), 0.5); \tag{6.1}$$

The relational information for a keypoint in GRM is represented as $\{FDL, D, w\}$, where FDL is the L2 subspace representation of a feature point with the graph center, D is the keypoint descriptor, and w is the weight (importance) of a feature. Furthermore, ICM encodes the holistic appearance using color histogram, and BDM encodes the spatio-temporal neighborhood local information for the pixels contained in the initialized bounding box. Table 6.1 summarizes the multiple appearance models with their respective feature description.

6.3.2 Graph similarity matching using GRM

With every new frame being processed by detecting and describing keypoints, our method tries to find a subgraph S in the frame that maximizes the similarity with the GRM by matching their keypoint descriptors. Let us denote an object GRM as G. For finding the center of the face, we will use the L2-subspace representation, FDL and its importance, w, associated with the matched keypoints to obtain the center. Hence, we are trying to find a subgraph S, which is *isomorphic* to G or to a subgraph of G. Thus, the maximum similarity between the two graphs can be represented as a function given by Equation 6.2 :

$$sim(G, S) = D(G) \sqcap_m D(S) \tag{6.2}$$

where, $\sqcap_m$ is the *bijection* that represents the keypoint matches, and D is the feature descriptors of the two graphs respectively. The total number of matched keypoint descriptors with D, at current frame, t, is given by N [1]. Now, we use this similarity knowledge to get the face center by using the relational information $FDL = [\Delta x, \Delta y]$, and matched keypoint, k in S. Thus, the face center given by a matched keypoints in S, can be represented using Equation 6.3.

$$x^t_{Center^k} = k_{x,y} + FDL \tag{6.3}$$

However, the subgraph may contain errors due to noise. Further, their structure cannot be determined in advance. Therefore, $x^t_{Center^k}$ is approximated using kernel responses denoted as φ. Two kernel functions are used for generating the response : Gaussian kernel, Φ_1, and Exponential kernel, Φ_2, respectively [2], and φ_k is represented by Equation 6.4 :

$$\varphi_k = \Phi_1(x^t_{Center^k}).\Phi_2(x^t_{Center^k} - x^{t-1}_{Center}).w \tag{6.4}$$

where w is the importance of a matched feature keypoint in G and x^{t-1}_{Center} is the face location in frame $t-1$. Now, all the N kernel responses are accumulated, i.e, they get overlapped. The face center location is obtained by analyzing the peak in the kernel response map, and is given by Equation 6.5 :

$$x^t_{Center} = \max_x \left(\sum_{k=1}^{N} \varphi_k(x) \right) \tag{6.5}$$

The obtained peak response is transformed back into the image coordinate system to obtain the face center location. As shown in Figure 6.2 the peak of the response (color coded as dark red), corresponds to the face center location. Hence, x^t_{Center} denotes the optimal solution for the face center target obtained by GRM model at frame t. While analyzing the kernel map, it is noted that the response is *anisotropic*, because of the different overlapping rates of the individual responses in the kernel map. This type of response proves highly beneficial for face localization by GRM from a regression perspective. In our method, during the kernel response generation, Φ_1 is centered at the face center location given by Equation 6.3, such that it gets the highest value. On the other hand, Φ_2 is highest when the face center given by the matched feature using Equation 6.3, is closer to the peak, x^t_{Center}. This helps to gain leverage over the short-term matched features in GRM that become relevant in generating kernel responses. As seen later in subsection 6.3.5, by analyzing the response for the features that are outputting correctly for the center, their influence in the kernel response map increases and reduces for others that are predicting wrongly or farther from the x^t_{Center}.

[1] [98] uses ratio test to eradicate matches higher than 0.8. In FaceTrack experiments 0.75 is used.

[2] The Gaussian kernel parameters are $\sigma = 6.0$, with a 5×5 filter size. The denominator, Θ of the Exponential kernel is taken as 8000.0.

6.3.3 Scale Adaptation and Computation of Appearance Similarity Scores

To adapt to the scale variation of the face, we use the same strategy, as used in [2]. The authors utilized pairwise distances between matched keypoints between consecutive frames to tackle scale change. Now for the output face location obtained using GRM, denoted as x^t_{Center}, face *candidates* are generated around it, to improve localization precision, since the center may get shifted due to face deformation or tracking noise. Apart from this, the second component of the framework, i.e. the face detector, outputs a bounding box for a detected face for frame t. The obtained bounding box from the detector is also considered as a face candidate.

Table 6.2 Computation of face appearance Similarity Scores in FaceTrack

Similarity Scores	Description
Keypoint Score	$K_{fc_i} = \frac{n}{N}$,
Color Score	$C_{fc_i} = \sqrt{\sum_{i=1}^{d}(ICM_{am} - ICM_{fc_i})^2}$,
Binary Descriptor Score	$B_{fc_i} = BDM_{am} \oplus BDM_{fc_i}$,

n, is the number of matched keypoint descriptors present in face candidate, fc,
N, is total number of matched keypoint descriptors of GRM that were matched at frame t,
d, is the feature dimension, *am* denotes the appearance template model for ICM and BDM,
$\oplus$ represents an operation.

Next, for all the face candidates, the ICM and BDM models are first computed, and are matched for similarity. Table 6.2 describes the formula for the computation of the respective similarity scores. The ICM model is compared using the norm *L2* norm, and is called Color Score, C_{fc}. The BDM model is compared using *hamming* distance, and is called by Binary Descriptor Score, B_{fc}. Further, a Keypoint Score, K_{fc}, for the matched keypoints in GRM, lying inside inside a face candidate box is computed. The features are normalized and transformed to the range $[0, 1]$. All the similarity scores, K_{fc}, B_{fc}, and C_{fc}, associated with fc, are used for obtaining the best face box by using a weighted score-level fusion strategy, as we will see later in subsection 6.3.4.

Algorithm 1 **FaceTrack Algorithm**

```
for all keypoints matched in subgraph, S do
    obtain face location using Equation 6.5 and generate face candidates
    adapt scale using pairwise keypoint distance
    for all face candidates from GRM and face detector, at frame $t$ do
        compute Similarity Scores, refer Table 6.2
        compute variance of Similarity Scores
        compute $FS_{fc_i}$ using Equation 6.6
    end for
end for
best face box as face candidate with max $FS_{fc_i}$
update appearance models using Algorithm 2
```

6.3.4 Face Localization using Weighted Score-level Fusion Strategy

For choosing the best candidate as the final output by the face tracking framework, we propose a strategy that combines the fusion of all the similarity scores (See Table 6.2), with weights based on their variance between two consecutive frames, such that the similarity score having the largest variance, gets the largest weight. If we just take the similarity score into account without its weighted variance, the fusion score might get higher for a candidate (e.g. distractor), even though it is not the face of interest which is required to be tracked. Moreover, the information from each appearance model are uncorrelated and by following this strategy, the contributions from each component can be utilized for maximum similarity. Thus, the best face candidate should maximize the following Equation 6.6.

$$FS_{fc_i} = p \cdot K_{fc_i} + q \cdot C_{fc_i} + r \cdot B_{fc_i} \tag{6.6}$$

where, p, q and r represent the weights assigned to the similarity scores, based on their variance ranking. The weights are assigned such that if $var(K_{fc_i}) > var(C_{fc_i}) > var(B_{fc_i})$, then p gets multiplied with K_{fc_i}, q with C_{fc_i}, and r with B_{fc_i}, respectively. The ranking helps to determine the dominant similarity score in a face candidate, and fusion helps to choose the best candidate that maximizes all the similarity scores. Algorithm 1 summarizes the proposed tracking framework.

6.3.5 Occlusion Detection, Tracking Control and Update Strategy

We consider two complementary aspects in tracking, robustness and stability, by long-term and short-term update. Long-term update are performed during the whole tracking duration for all the keypoint features, k_i, collected for GRM model at frame t, by adapting their weights using Equation 6.7 :

$$w_{k_i}^{t+1} = \begin{cases} (1-\tau)w_{k_i}^{t} + \tau \cdot \theta(l), & \text{if } k_i \epsilon N, \\ (1-\tau)w_{k_i}^{t}, & \text{otherwise} \end{cases} \tag{6.7}$$

where τ is *learning rate*. The value of $\theta(l)$ increases with a keypoint prediction closer to the x^t_{Center} and is obtained using Equation 6.8 as :

$$\theta(l) = \max((1 - |\eta \cdot l|), 0.0); \tag{6.8}$$

where l is the L2 distance between the center location given by the matched feature keypoint using its relational information, and the center obtained by analyzing the response in the kernel map, x^t_{Center}. On the other hand, tracking control is done by analyzing the center response given by a matched keypoint. It is done to avoid potential tracking failures. For example, for a given frame t, if a matched keypoint outputs a center farther from the center ($x^t_{Center^k}$) in frame $t-1$, then its influence in the kernel response map for future frames get reduced using exponential kernel function, Φ_2 (used in Equation 6.4). It is given by the following Equation 6.9 :

$$\Phi_2 \propto \exp \frac{-(x^t_{Center^k} - x^{t-1}_{Center})}{\Theta} \tag{6.9}$$

By *controlling* this, potential tracking drift failures can be avoided, which in turn gives the proposed method stability along with its robustness towards face appearance changes.

When the similarity between graphs cannot be established in a frame (i.e. no subgraph S can be matched), we consider this scenario as an occlusion detection, and perform short-term update by partially (or fully)[3] updating the ICM and BDM model respectively.

During this scenario, the ICM and BDM models help to localize the face target, since the similarity scores of these models will dominate for the best face candidate. New features are added to GRM when the ICM and BDM similarity matching score for the face output template is above α and β, respectively. Features having weights lower than γ, are removed from GRM.

Thus, by following this control and update strategy, the different appearance models complement each other during different tracking scenarios. Algorithm 2 summarizes the update strategy of the proposed face tracking framework.

[3] partial update : by *replacing* 12.5% of the face appearance features in ICM model and 10% of the face appearance features in BDM model respectively, full update : by *replacing* 100 % of the ICM and BDM model

Algorithm 2 **FaceTrack occlusion detection, control & update strategy**

```
for keypoints, $k_i$, in GRM do
    Long-term update using Equation 6.7
    Tracking control using Equation 6.9
    if $w_{k_i} < \gamma$ then
        Remove $k_i$ from GRM
    end if
    if (N == 0) then
        Occlusion detected
        if (appearance templates size != best face box size) then
            partial update of ICM & BDM models
        else
            full update of ICM & BDM models
        end if
    end if
end for
if $K_{fc_i} > \alpha$ and $B_{fc_i} > \beta$ then
    Add new keypoints in GRM
    full update of ICM & BDM models
end if
```

6.4 Experimental Evaluation

The proposed method is validated on OTB benchmark [68] for One-Pass Evaluation (OPE). The selected state-of-the-art trackers used for comparison are : Struck [79], TLD [42], KCF[66], MIL [5], CMT [80], TUNA [2] and Boosting [73]. 15 video sequences from the benchmark containing faces are chosen for evaluation. These video sequences display several challenges that are encountered during tracking a face in a video sequence : occlusion (OCC), fast motion (FM), illumination variation (IV), scale variation (SV), motion blur (MB), in-plane-rotation (IPR), out-of-plane rotation (OPR), background clutter (BC), out-of-view (OV) and deformation (DEF). Table 6.3 shows the distributions of attributes for the 15 face video sequences with different challenges.

6.4.1 Evaluation Metrics

The benchmark is evaluated on two performance measures : precision and success. *Precision* is measured as the distance between the centers of a bounding box outputted by the tracker and the corresponding ground truth bounding box. The precision plot shows the percentage of frames whose center localization output are within a given threshold distance. *Success* is

Table 6.3 Distribution of attributes of the 15 video sequences : Motion Blur (MB), Fast Motion (FM), Background Clutter (BC), Deformation (DEF), Illumination Variation (IV), In-plane Rotation (IPR), Occlusion (OCC), Out-of-plane-Rotation (OPR), Out-of-View (OV), Scale Variation (SV).

Video Attributes	MB	FM	BC	DEF	IV	IPR	OCC	OPR	OV	SV
Total Number	5	5	4	4	5	12	7	13	1	10

measured as the intersection over union of pixels bounding box outputted by the tracker with the ground truth bounding box. The success plot shows the percentage of frames with their overlap score higher than a set of all the given thresholds, t, such that $t \epsilon [0, 1]$.

For our experiments, we test all the trackers by initializing them in two ways : (1) Ground truth initialization, (2) Automatic initialization using a face detector [16] [4]. All the selected trackers for comparison are implemented in the OpenCV 3.1.0 library except Struck [5], CMT [6] and TUNA [7], for which the code is provided online by the authors. The trackers are evaluated using the default parameters provided in their respective research papers. The proposed FaceTrack is tested on machine with configuration as Intel Core i7 @ 3.40GHz, 16GB RAM and is implemented in C++.For evaluation, the parameters of FaceTrack are : $\alpha = 0.23$, $\beta = 0.1$, $\gamma = 0.1$, $p = 0.15$, $q = 0.1$, $r = 0.1$, $\tau = 0.9$, and $\eta = 0.005$. They are fixed for all the experiments. Face tracking results can be found at http://step.polymtl.ca/∼Tanushri/FaceTrack/.

6.4.2 Comparison to state of the art

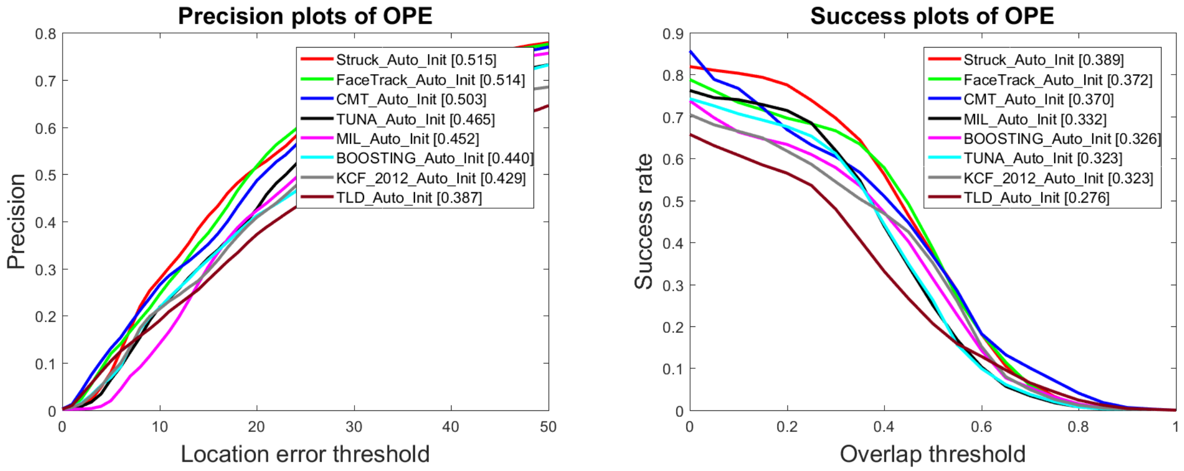
FaceTrack shows strong performance when initialized using ground truth : precision, 0.603 and success, 0.425, respectively. Furthermore, it outperforms and ranks *second* in overall performance when initialized automatically (Refer Figure 6.3). This performance with automatic initilization showcases that FaceTrack is comparatively less affected by initialization. The robustness of FaceTrack can be attributed to its robust initialization strategy using *isotropy* in which all features are not given equal importance. The keypoint features that get matched with GRM appearance model output a subgraph, containing an estimate of a region that contains the target face. However, for finer estimation for face location, the kernel responses of the matched features are summed. This response is *anisotropic*, which efficiently determines precise face location, as it corresponds to maximum value of the cumulative overlapped responses. The response is guided through long-term update of features that are analyzed

[4] Any other face detector can be used for initialization purpose.

[5] https ://github.com/samhare/struck

[6] https ://github.com/gnebehay/CppMT

[7] https ://github.com/sinbycos/TUNA



(a)

(b)

Figure 6.3 FaceTrack performance in real-world scenarios when initialized automatically : (a) overall precision and (b) overall success. (Best viewed when zoomed in.)

using multiple kernel functions (Refer subsections 6.3.2 and 6.3.5 for details). Further, for finer precision, face candidates are generated around this location and the bounding box given by the face detector is also considered a candidate. Finally, the weighted score-level fusion score helps to decide the best face candidate.

The uniqueness of GRM lies in its design, as it helps in tracking a specific face. The approximation of FDL using the Gaussian kernel helps to tackle face deformation, which happens very often during face tracking. During deformation, the keypoint feature can move by a pixel which can result in error. Thus, approximating the response using a Gaussian kernel compensates for this error, and in turn for face deformation. Even during heavy occlusion, in-plane rotation, the short-term updates help to locate the target face as the appearance matching can still be established with the aid of multiple appearance models during such scenarios. On the other hand, during drastic appearance changes like scale change, the face detector tackles it even if the some of the keypoint features in GRM may fail to get matched. However, in cases when no appearance matching can be established and the face detector also fails to detect a face, then the face location is not updated until the face appearance matching starts establishing again. However, it might be possible that the face detector outputs false positives. In addition, since it does not use any spatio-temporal information of the target face from the previous frame, its detection might be for a distractor. Therefore, in this case, the face candidates generated around the localized face by the GRM model will dominate in localizing face since their similarity score of appearance will be higher, thus, avoiding wrong face localization.

Table 6.4 Comparison of FaceTrack with the state-of-the-art trackers on 15 video sequences with various challenges. The bold text showcases the trackers most affected towards initialization.

Algorithm		*Precision*			*Success*	
	GT Init	*Auto Init*	*%Relative change in Precision*	*GT Init*	*Auto Init*	*%Relative change in Success*
FaceTrack (Proposed)	0.603	0.514	14.76% ↓	0.425	0.372	12.47% ↓
Struck [79]	0.705	0.515	**26.95**% ↓	0.543	0.389	**28.36**% ↓
TLD [42]	0.432	0.387	10.42%↓	0.335	0.276	17.61%↓
KCF [66]	0.623	0.429	**31.14**%↓	0.478	0.323	**32.43**%↓
MIL [5]	0.496	0.452	8.87% ↓	0.383	0.332	13.32% ↓
Boosting [73]	0.520	0.440	15.38% ↓	0.419	0.326	22.20% ↓
CMT [80]	0.649	0.503	**22.50**% ↓	0.506	0.370	26.88%↓
TUNA [2]	0.598	0.465	22.24% ↓	0.475	0.323	**32.00**% ↓

It is interesting to note that the performance results become more interesting when FaceTrack is initialized automatically and ranks just after Struck by a very minute margin. It can be noted in Table 6.4 that the percentage drop in terms of performance is on the higher side, almost double for Struck, KCF, CMT and TUNA as compared to FaceTrack indicating that FaceTrack is less affected by the initialization as it gets re-initialized periodically when the face candidate sample is chosen. Moreover, the proposed occlusion detection, tracking control and update strategy that helps FaceTrack robust towards appearance changes but at the same time be less affected from distractions, thus outputting stable results. In addition, the use of the face detector aids in drastic appearance changes of target face. This also indicates that the model update which involves addition of new features and deletion of bad features that are not predicting for center in GRM, partial and full update of ICM and BDM models is most of the time happening correctly. An untimely update might result in corrupting the appearance models, and the tracker might fail. The next subsection gives detailed attribute-wise analysis of FaceTrack and how it is able to tackle various tracking challenges.

6.4.3 Attribute-wise Analysis

FaceTrack outperforms several state-of-the-art trackers by ranking first or second on almost all the tracking nuisances when initialized automatically (See Figure 6.4, 6.5, 6.6). The following paragraph details the analysis.

Scale variation and rotation : Together with the keypoint scale adaptation strategy from [2], and scale and aspect ratio adaptation from a face detector, the tracker performs well in tackling scale variation of the face, which is a common phenomena during object tracking. As long as the face remains partially or fully visible during in plane rotation and out-of-plane rotation, all the appearance models namely, GRM, ICM, and the BDM models contribute in face localization by maximizing the fusion score for all the face candidates. However, during out-of-plane rotation, GRM might be hidden and may not able to localize the face. On the other hand, because of the control and update strategy of our framework, the ICM and BDM templates get partially or fully updated (refer Algorithm 2). Hence, during this time, ICM and BDM similarity score will dominate in maximizing the fusion score of face candidates.

Fast motion and motion blur : FaceTrack effectively deals with fast motion and motion blur during tracking by maximizing graph similarity in the whole frame. Further, having a face detector helps to find target during motion blur, since it does not suffer from the problem of drift due to its image independent searching principle (no spatio-temporal information is used).

Background clutter : The distinct appearance model GRM tackles the complex background

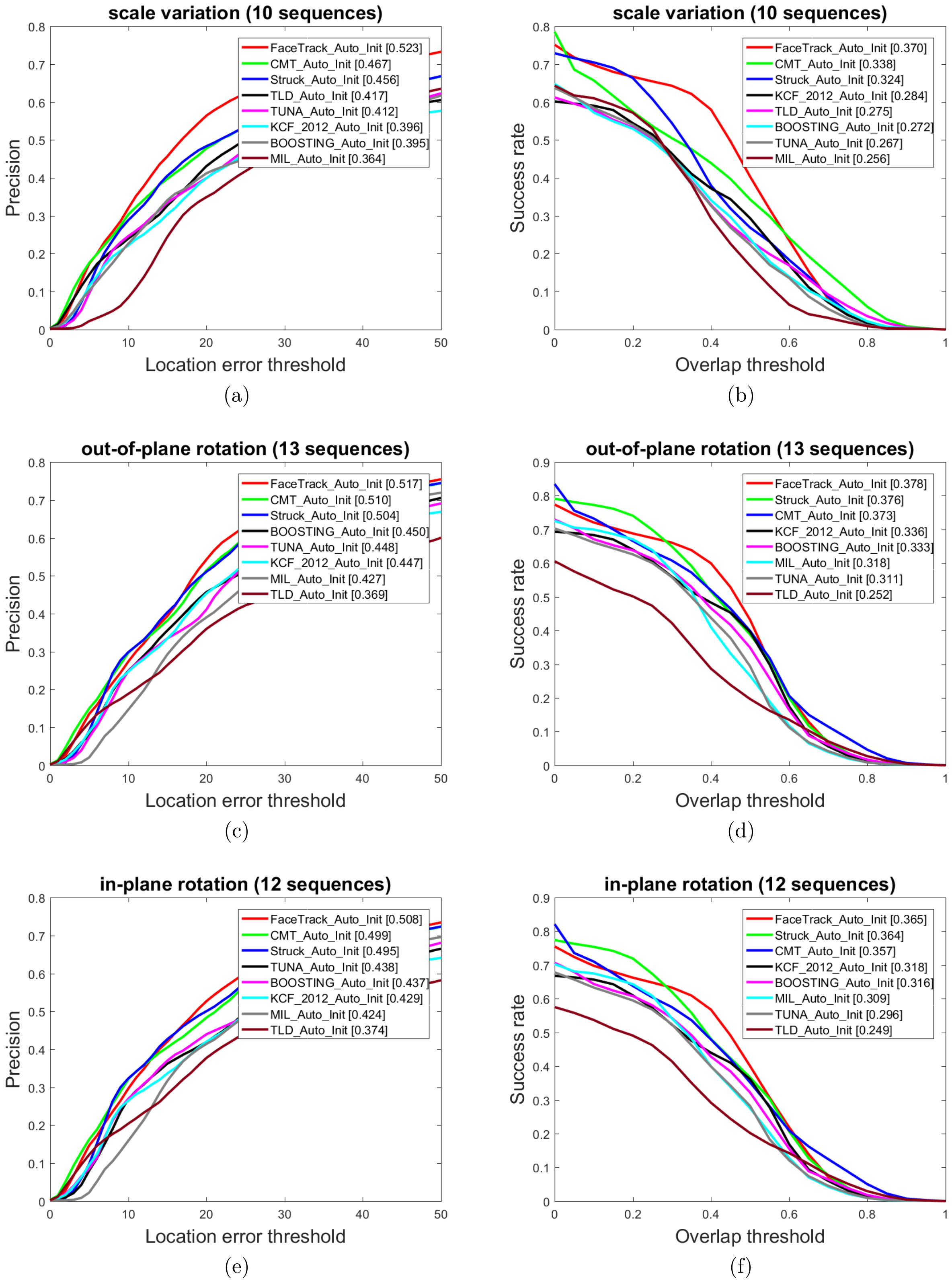


Figure 6.4 FaceTrack performance on video attributes : (a) & (b) scale variation, (c) & (d) out-of-plane-rotation, (e) & (f) in-plane-rotation (Best viewed when zoomed in.)

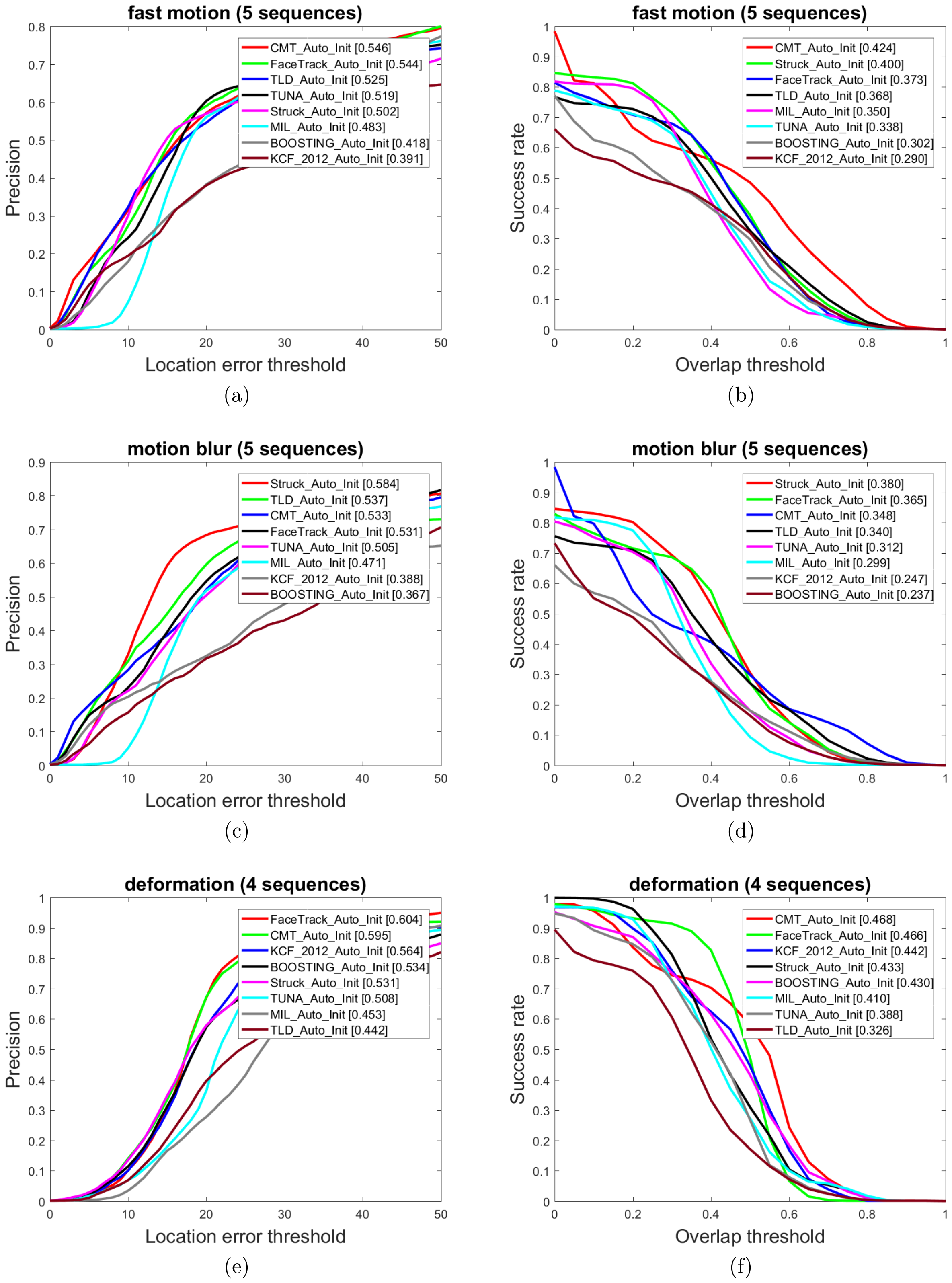
and helps to identify the face during background clutter. During such scenarios, it becomes difficult to discriminate the face target from the background. But thanks to the L2-subspace based GRM appearance model that preserves the internal structural representation of the target face by assigning importance to the features that are memorized for long duration. Hence, the incremental learning of the model helps to capture the appearance representation and thus making it easier to track a face.

Illumination variation and Occlusion : When the target face undergoes severe illumination change in sequences, most of the methods tend to drift towards the cluttered background or cannot adapt to the scale change that occurs during this time. In addition, during this time and also during occlusion, the appearance of the target face changes drastically. Therefore, the GRM model is unable to localize the target face. Therefore, ICM and BDM, can be utilized for a short-term reference model for appearance matching. These models get updated frequently for short-term according to the update and control strategy in the proposed tracking framework. In addition, the face detector facilitates face localization and adaption of scale change during such drastic appearance change.

Deformation : The proposed tracker is able to handle object deformation very well in sequences. This is because during deformation, the FDL associated with some of the keypoints in GRM may differ in length as the keypoints may get shifted from their original location. But the summing of the various kernel responses generated using multiple kernels in the response map, compensates for this error. Moreover, the face detector aids in reinitialization of the tracker in case the tracker drifts away from the target face.

Failure cases : The tracker may sometime loose track of a face in videos having drastic appearance change. In addition, it might be possible that the face detector is unable to detect the target face and output false positives. Thus, during this scenario, the similarity of face appearance cannot be established, due to which the face location might not get updated. Hence, during such a scenario the face might not get tracked. But, if a correct face detection for the target face can be obtained, then tracker will get re-initialized and will resume tracking.

In summary, FaceTrack is able tackle the various tracking nuisances by utilizing the different components built in its algorithm. The next subsection presents an ablation analysis of FaceTrack.



(a)

(b)

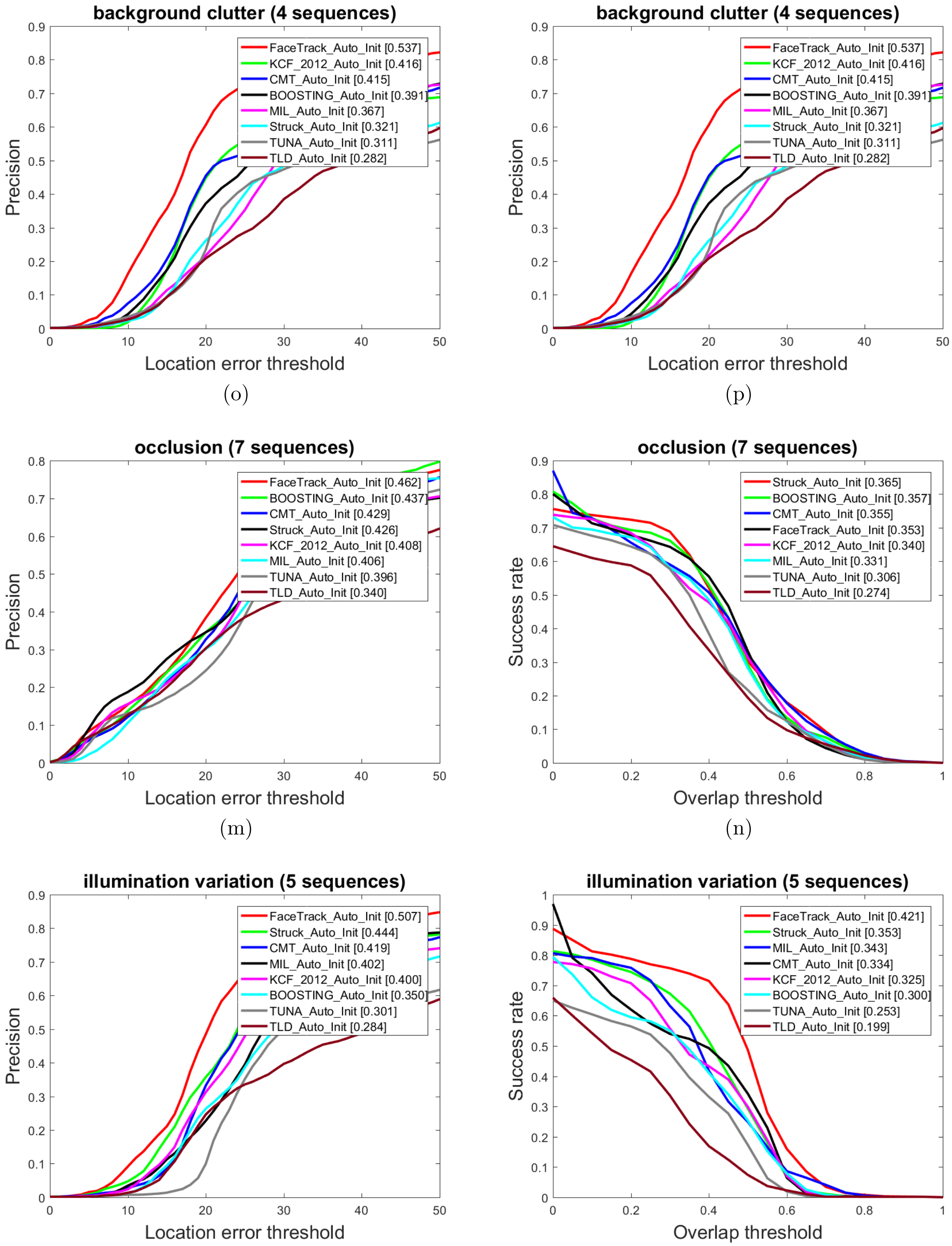
(c)

(d)

(e)

(f)

Figure 6.5 FaceTrack performance on video attributes : (a) & (b) fast motion, (c) & (d) motion blur, (e) & (f) deformation. (Best viewed when zoomed in.)

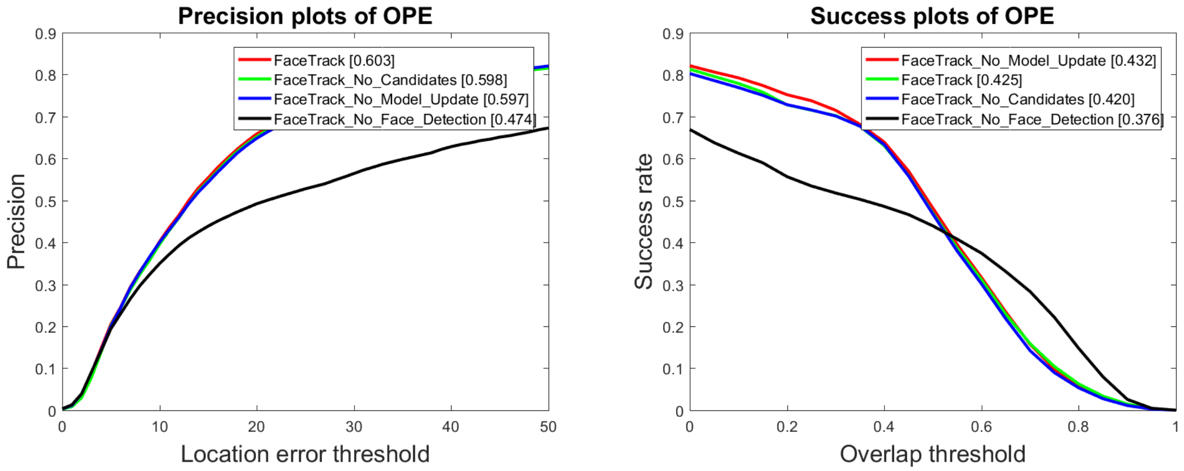


(o) (p)

(m) (n)

(q) (r)

Figure 6.6 FaceTrack performance on video attributes : (a) & (b) background clutter, (c) & (d) occlusion, (e) & (f) illumination variation. (Best viewed when zoomed in.)



(a) (b)

Figure 6.7 Ablation analysis of FaceTrack (a) Precision for One-Pass Evaluation (b) Success for One-Pass Evaluation. (Best viewed when zoomed in.)

6.4.4 Ablation Analysis

To demonstrate the effectiveness of each component in the FaceTrack tracking framework, we eliminate in turn a component from it. For e.g. removing face detector, or removing the generation of face candidates or by not performing any appearance model updates for GRM, ICM and BDM respectively. It can be seen in Figure 6.7 that removing the face candidates from FaceTrack reduces its performance, which confirms our hypothesis that face candidates help in better face localization. Removing the face detector from FaceTrack results in performance loss indicating that face detector helps to tackle the drastic appearance changes during face tracking. By *not* performing model update (both partial and full) of ICM and BDM models and not adding/deleting connections from GRM, but only doing updates using Equation 6.7 and control using Equation 6.8 respectively, the performance of FaceTrack falls in precision but improves in success. This can be attributed to the fact that it is very challenging to perform correct appearance model update at all times during the online tracking process in the *absence* of ground truth, which always involves a risk. On the other hand, if updates are not performed at all, then the face tracker might not be able to cope up with the changing appearance of face and eventually loose track. Thus, keeping all this in mind, it can be seen from Figure 6.7 that even without updating ICM and BDM appearance models, FaceTrack is successfully able to track a face almost 60% of the time, showcasing its robustness towards appearance change by adapting weights of keypoints present in the GRM model, and stability through its control strategy, tracking-by-detection using face detector, along with the weighted score-level fusion strategy for precise face target localization. Thus, all components play an important role in robust face tracking.

6.4.5 Time Complexity Analysis

By referring Algorithm 1, it can be approximated that the time complexity of FaceTrack is $\approx O(n^2)$. FaceTrack estimates the face location as the face candidate having the highest fusion score. The similarity score of all the face candidates is obtained by maximizing the similarity of appearance models : GRM, ICM and BDM respectively. Please note that the matching of GRM model also accounts for keypoint extraction between two frames, matching keypoint descriptors between two frames and then finding the maximum in the kernel response map. The video sequences contain different frame resolution. FaceTrack runs with an average of 2 frames-per-second computed over 15 video sequences on an Intel Core i7 with a 3.40 GHz clock and 16GB RAM (with KeyPoint descriptor matching between two frames on NVIDIA GeForce GTX 560 graphic card).

6.5 Conclusion

In this paper, FaceTrack is proposed. It utilizes multiple appearance models for robust face tracking. The proposed multiple appearance models account for the *temporal* (both long-term and short-term) appearance change of a face during tracking. FaceTrack jointly takes the advantage of the multiple appearance models by matching them effectively during different tracking scenarios to facilitate tracking. The incremental graph relational learning using the long-term update of face appearance features help to localize the face by finding an *isomorphic* subgraph. The matched subgraph is approximated using multiple kernel functions in a kernel response map. The multiple kernels help to tackle face deformation and potential face tracking failures. In addition, the approximation also encodes error and eradicates its effect for precise face target location, by determining a non-linear decision boundary in the anisotropic kernel response map. In addition, the face detector helps to localize the face during drastic short-term appearance change and reinitialization of FaceTrack. Furthermore, for precise face location, face candidates are generated and the final face location is chosen as the candidate having the highest fusion score. Extensive experiments showcase the effectiveness of each component of the proposed face tracking framework for many tracking real-world unconstrained tracking nuisances in terms of accuracy, robustness, adaptiveness and tracking stability. In conclusion, it is essential that the face tracker should robustly adapt to appearance changes, and at the same time should output stable tracking results in spite of distractions which cannot be controlled in real-world scenarios.

6.6 Acknowledgement

This work was supported in part by FRQ-NT team grant #167442 and by REPARTI (Regroupement pour l'étude des environnements partagés intelligents répartis) FRQ-NT strategic cluster.

CHAPTER 7 GENERAL DISCUSSION

In this chapter, we revisit the objectives mentioned in the thesis and discuss how our research contributions helped in achieving those. We discuss here the key findings and observations of the research that were both advantageous and disadvantageous to object tracking. In addition, we identified the research impact and applications of our work in the domain of computer vision engineering and machine learning.

7.1 From identifying tracking challenges to developing a robust Model-free tracker : CTSE, [1]

In our research, we first worked on identifying the challenges that occur in visual object tracking applied to the case of a face. During that time it was observed that no matter what the object to be tracked, the tracking nuisances are similar. Hence, the idea to pursue a model that could be learned in one-shot (learning from a single sample) came into realization. In visual object tracking, model-free trackers outline the problem of tracking in a similar manner. The problem of tracking is challenging because it not only requires precision but also requires speed, i.e., there cannot be much delay in the processing time between the input frame and the tracking output. Designing a tracking method that consumes a lot of time in processing of video frames, will very much destroy the purpose of tracking in video surveillance. Moreover, it was realized that features that can be extracted on a variety of objects that are particularly distinct from the background will facilitate tracking. Hence, the first contribution is about tracking different kind of objects from a single sample with the help of keypoints. SIFT keypoints are used, as they encode local neighborhood information of a keypoint using feature descriptors. The keypoints are robust to illumination, and are scale invariant. The features in the appearance model adapt themselves online, because it was noticed that if there is no incremental learning of features, then the tracker was failing in some specific scenarios. The dataset on which the tests were experimented contained similar objects in the surrounding, which was another tracking challenge. Thus, came the idea of using a structure that can be associated with a feature. The hypothesis is that during tracking even though the features might change abruptly, the structure associated with the keypoint will not undergo a very rapid change. Hence, together the features with a structure tackles the distractor nuisance and the experiments confirmed the hypothesis. In addition, it was realized that context around the target object can be beneficial for tracking. To measure its effectiveness, experiments were conducted with and without the context region. The experiments proved that adding context will help in tackling objects having certain characteristics only in specific scenarios. For example, the context from the neck region can be considered for tracking a face. It has similar motion to the face of the person. But if the context region also gets occluded, then it is not beneficial for tracking the face anymore. Therefore, having a context may not contribute in robust tracking all the time. Furthermore, in order to localize the object correctly, a strategy is required to deal with the inherent tracking noise during the process of tracking in video sequences, which is an important observation to be noted. This tracking noise is difficult to remove ; it is not possible to have prior knowledge of the noise which can be formulated in terms of a probability distribution. Hence, a method is proposed, called voting by keypoints, that aids in reducing its influence on target localization. The experiments show that the proposed tracker is performing at the top on challenging video dataset [1].

7.2 Adaptation of Model-free tracker towards scale variation and long-term tracking : TUNA, [2]

During this phase, the work focused on developing the aforementioned model-free tracker to tackle scale change of the target object. It was noticed that the model-free tracker was failing in sequences with scale variation of the object. This pointed out that the structure associated with a feature should also be utilized for gauging the scale change of the object between a number of frames, since the scale of the object does not vary much between two consecutive frames. This strategy enabled to adapt to the scale variation of the target object tested for 51 sequences. But in case of no keypoint matches between frames, the scale will not get adapted. Since the dataset used for experimentation contained a wide variety of objects, ranging from a *motorcycle* to a *deer*, the experimentation on the dataset helped to identify the strengths and shortcomings of the current tracker. It was realized that only having a single cue (only tracking with keypoints) cannot effectively tackle tracking nuisances. Therefore, multiple cues of the target object were utilized as features, which made it robust towards distractors, low-resolution. The experimentation proved that the tracker can be used to track for a wide range of objects. Justification for choosing multiple cues is described previously. Moreover, it was found that the search region for feature matching should not be too limited as it cannot tackle object with large motion, instead the feature matching can be done in the complete frame. The purpose of doing this was to track objects with large motion, for e.g. a boy jumping or skipping. This is an another finding to be noted. Further, a strategy is proposed to add new features and delete non-performing features to the appearance model, which helps in tracking the target object over long periods in video sequences. Tracking over

[1] https ://bitbucket.org/tanushri/ctse

long-periods shows the robustness of the appearance models and appearance update strategy. This was realized during the research and development of this tracker [2].

7.3 Face Tracking using multiple appearance models and a face detector : FaceTrack, [3]

In the final research phase, the work focused on developing a face tracker for unconstrained video sequences by building a framework that utilizes multiple appearance models : GRM, ICM and BDM respectively. The models deal with numerous tracking nuisances. In addition, the face detector handles abrupt motion changes and scale of the target face due to its inbuilt still image face detection module, but cannot handle occlusions very well. On the other hand, the appearance models can handle partial occlusions, but might drift during drastic appearance change of the face. But, due to the shortcomings of both the components, neither detection nor tracking *alone* can solve the complex challenges that occur while tracking faces. This came into picture during the experiments. It is possible that even after localization by the GRM appearance model, the face localization may not be precise due to face deformation and tracking noise. Hence, a strategy is devised that generates face candidates around the tracking location. But, then to decide the best face location among them is a crucial process, and requires a decision strategy. This is another key finding to be noted. Thus, a weighted score-level fusion approach is proposed for selecting the best final output, which utilize the variance of the similarity scores of the multiple appearance models that are weighted by their similarity variance between consecutive frames. By doing this, the candidate which maximizes the fusion of these similarity scores is chosen as face location. Since, if we just take the similarity score into account without its weighted variance, the fusion score might get higher for a candidate (e.g. distractor), even though it is not the face of interest which is required to be tracked. This is an important observation to be noted. Apart from this, a tracking control and update strategy is proposed to maintain a short-term and long-term appearance of the target face during tracking. By following this strategy, the target face can be tracked during severe appearance change, without dithering the performance of the face tracker [3].

7.4 Research Impact

Our work on visual object tracking led to development of three robust trackers namely CTSE, TUNA and FaceTrack, and can be used to track a specific as well as non-specific objects.

[2] https ://bitbucket.org/tanushri/tuna

[3] https ://bitbucket.org/tanushri/facetrack

The tracker source codes are open-sourced, and are implemented in C++ using OpenCV[4] computer vision library for easy adaptation in other projects. For example, for tracking cars, the same components of the tracker can be used just by changing the detector. The tracker can also be used in robotic applications besides video surveillance. In addition, different features can be combined and still use the weighted score-level fusion score.

The research in object tracking is moving rapidly towards deep learning features and will continue to progress in this direction for quite sometime, which marks a major shift from the traditional tracking techniques to artificial intelligence. Thus, deep features can also be used by keeping the proposed tracking methodology.

7.5 Limitations

Even though the tracking performance by the trackers is encouraging for video sequences having a variety of attributes, some limitations in our research still exist. The main is the improvement of the precision of the tracker by adding an object detector. Using an offline trained detector cannot guarantee to detect the object at every frame. During offline training, thousands of samples must be collected for the specific object to be tracked, which is other aspect to be considered while training the detector model. In contrast, online training of the detector during tracking can be done by providing samples, but the decision boundary for these samples should not be binary, instead a non-linear one. Having a hard decision boundary might reject samples that might be useful for tracking an object. Finally, in tracking, precision and success go hand in hand. Hence, by improving the precision of the tracker, a large improvement in the success can also be witnessed.

[4] http ://www.opencv.org/

CHAPTER 8 CONCLUSION AND RECOMMENDATIONS

8.1 Concluding Remarks

In this research project, the problems that occur while tracking an object in video sequences were studied in detail, and to tackle them effectively, some solutions were proposed that can be generalized for tracking a variety of objects in real-world unconstrained video sequences.

Assessing the tracking performance on an extensive evaluation benchmark provided the opportunity to conduct an ablation analysis on the performance of the proposed tracking system on videos having different capture conditions and different attributes. The tracking parameters are kept the same for the whole dataset. This helped to identify the aspects that can be improved in the future work.

In our first phase of the research, the objective was to focus on the features that are robust and that can be extracted in a shorter period of time. Another objective was to initialize tracking right away from a single example, without requiring sophisticated training modules. Thus, *CTSE* was developed with the purpose of tracking different kinds of objects, like a model-free tracker and not limited to a specific object. The first contribution is that the use of features with structure and context that have motion correlation with the target object center may be combined for robust tracking. Second, for target localization, a voting-by-keypoints strategy is proposed. Finally, the third contribution is to estimate the quality of features by keeping a structural configuration of the target that is updated online for better target localization and robustness to the appearance changes of the object.

In the second phase of the research, a tracker called *TUNA* was proposed towards tackling object deformation and scale changes of the object. Here, the anchor point feature appearance model is proposed, which is a distinct appearance representation of the target object. This model tackles partial occlusion and distractors in the tracking scenario. The second contribution is determining the correct time to update the anchor point appearance model, as a wrong update might result in tracking drift and ultimately failure. Finally, to preserve robust features for tracking, the short-term and long-term consistency of a feature is determined to keep good features for target center prediction and removing the ones that are not predicting correctly.

In the final phase of the research, all the advantages of the aforementioned trackers are combined into one tracker called *FaceTrack*, with the motive of tracking a face in an unconstrained scenario. The work focused on addressing the problem of face appearance matching across different tracking challenges. A multiple appearance model with graph relational information is proposed. The appearance of a face is represented using a L2-subspace and is used as a relational information together with its isotropic weights and feature descriptor. Matching this graph across frames helps to find and localize a face. When this graph cannot be matched, occlusion is detected and other appearance models facilitate face tracking. Further, the graph matching is approximated using multiple kernels that helps to localize the face by generating an anisotropic response in the kernel map. This helps to arrive at a non-linear decision boundary for finding the target face. Further, to eradicate the effect of noise and face deformation, the face localization is achieved by analyzing the peak response in the kernel map. In addition, higher dimensional face data is encoded into global (color histogram), *spatio-temporal* local (binary descriptor and invariant keypoint descriptor) appearance models to effectively deal with tracking nuisances. In the proposed face tracking framework, an offline-trained face detector is used to adapting scale changes of the face. To decide the best face location, a weighted *fusion* strategy is proposed. Furthermore, tracking control and update strategy is proposed to identify the timely update of the proposed appearance models for robust and stable tracking.

8.2 Recommendations for Future Research

Based on our research experiences, we recommend future researchers to focus on object tracking challenges like low-resolution and long duration object tracking. This is because, in real-world scenario, the resolution of the surveillance video is low. Moreover, it is required that the tracking should continue for long duration and does not drift away from the target during such situation. The precision of the tracker can be improved by emphasizing on selection of samples for online *training*. It is essential that the face detector should be trained with these samples, during the process of tracking, as the appearance of the target changes quite often.

REFERENCES

[1] T. Chakravorty, G.-A. Bilodeau, and E. Granger. Contextual object tracker with structure encoding. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4937–4941, Sept 2015.

[2] T. Chakravorty, G.-A. Bilodeau, and E. Granger. Tracking using Numerous Anchor points. *ArXiv e-prints*, February 2017.

[3] T. Chakravorty, G.-A. Bilodeau, and E. Granger. Robust Face Tracking using Multiple Appearance Models and Graph Relational Learning. *ArXiv e-prints*, June 2017.

[4] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *CVPR Workshops*, pages 74–81, June 2011.

[5] B. Babenko, M. H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8) :1619–1632, Aug 2011.

[6] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR ’98, pages 232–, Washington, DC, USA, 1998. IEEE Computer Society.

[7] Junseok Kwon and Kyoung Mu Lee. Tracking of abrupt motion using wang-landau monte carlo estimation. In *ECCV (1)*, pages 387–400, 2008.

[8] Chakravorty T., Bilodeau G.-A, and Eric Granger. Automatic image registration in infrared-visible videos using polygon vertices. *CoRR*, abs/1403.4232, 2014.

[9] Xue Mei and Haibin Ling. Robust visual tracking using l1 minimization. In *IEEE 12th International Conference on Computer Vision,*, pages 1436–1443, Sept 2009.

[10] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2) :137–154, May 2004.

[11] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, pages 3074–3082, Washington, DC, USA, 2015. IEEE Computer Society.

[12] Miguel De la Torre, Eric Granger, Robert Sabourin, and Dmitry O. Gorodnichy. Adaptive skew-sensitive ensembles for face recognition in video surveillance. *Pattern Recognition*, 48(11) :3385 – 3406, 2015.

[13] M. A. A. Dewan, E. Granger, R. Sabourin, G. L. Marcialis, and F. Roli. Video face recognition from a single still image using an adaptive appearance model tracker. In *IEEE Symposium Series on Computational Intelligence*, pages 196–202, Dec 2015.

[14] Saman Bashbaghi, Eric Granger, Robert Sabourin, and Guillaume-Alexandre Bilodeau. Robust watch-list screening using dynamic ensembles of svms based on multiple face representations. *Machine Vision and Applications*, 28(1) :219–241, 2017.

[15] Xu Yan, Xuqing Wu, IoannisA. Kakadiaris, and ShishirK. Shah. To track or to detect ? an ensemble framework for optimal selection. In *ECCV*, Lecture Notes in Computer Science, pages 594–607. Springer Berlin Heidelberg, 2012.

[16] Shengcai Liao, Anil K. Jain, and Stan Z. Li. A fast and accurate unconstrained face detector. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2) :211–223, February 2016.

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9) :1627–1645, Sept 2010.

[18] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6) :810–815, June 2004.

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

[20] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.*, 4(4) :58 :1–58 :48, October 2013.

[21] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *ACM Comput. Surv.*, 38(4), December 2006.

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, November 2004.

[23] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[24] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure : Center surround extremas for realtime feature detection and matching. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2008.

[25] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, pages 1508–1515, Washington, DC, USA, 2005. IEEE Computer Society.

[26] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3) :346–359, June 2008.

[27] P. Vandergheynst, R. Ortiz, and A. Alahi. Freak : Fast retina keypoint. *IEEE Conference on Computer Vision and Pattern Recognition*, 0 :510–517, 2012.

[28] S.L. Happy, A. George, and A. Routray. A real time facial expression classification system using local binary patterns. In *International Conference on Intelligent Human Computer Interaction (IHCI)*, pages 1–5, 2012.

[29] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief : binary robust independent elementary features. In *Proceedings of the European conference on Computer vision*, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.

[30] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk : Binary robust invariant scalable keypoints. *IEEE International Conference on Computer Vision*, 0 :2548–2555, 2011.

[31] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR*, volume 2, pages 142–149 vol.2, 2000.

[32] David A. Forsyth and Jean Ponce. *Computer Vision : A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[33] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[34] Junseok Kwon and Kyoung Mu Lee. Highly nonrigid object tracking via patch-based dynamic appearance modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10) :2427–2441, 2013.

[35] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes : Active contour models. *Int. Journal Of Computer Vision*, 1(4) :321–331, 1988.

[36] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV, Washington, DC, USA, 2003. IEEE Computer Society.

[37] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 :888–905, 1997.

[38] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, ICCV, pages 555–, Washington, DC, USA, 1998. IEEE Computer Society.

[39] Samuele Salti, Andrea Cavallaro, and Luigi Di Stefano. Adaptive appearance modeling for video tracking : Survey and evaluation. *Trans. Img. Proc.*, 21(10) :4334–4348, October 2012.

[40] N.J. Gordon, D.J. Salmond, and A. F M Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, 140(2) :107–113, Apr 1993.

[41] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[42] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7) :1409–1422, July 2012.

[43] Ming-Hsuan Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images : a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :34–58, Jan 2002.

[44] M. Viola, Michael J. Jones, and Paul Viola. Fast multi-view face detection. In *Proc. of Computer Vision and Pattern Recognition*, 2003.

[45] Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiang Zhang, and Harry Shum. Statistical learning of multi-view face detection. In *Proceedings of the European Conference on Computer Vision*, ECCV, pages 67–81, London, UK, UK, 2002. Springer-Verlag.

[46] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP*, pages 900–903, 2002.

[47] Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, and Stan Z. Li. Face detection based on multi-block lbp representation. In *Proceedings of the International Conference on Advances in Biometrics*, ICB, pages 11–18, Berlin, Heidelberg, 2007. Springer-Verlag.

[48] Hongming Zhang, Wen Gao, Xilin Chen, and Debin Zhao. Object detection using spatial histogram features. *Image and Vision Computing*, 24(4) :327 – 341, 2006.

[49] Rainer Lienhart, Er Kuranov, and Vadim Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM Pattern Recognition Symposium*, pages 297–304, 2003.

[50] S. Charles Brubaker, Jianxin Wu, Jie Sun, Matthew D. Mullin, and James M. Rehg. On the design of cascades of boosted ensembles for face detection. *International Journal of Computer Vision*, 77(1) :65–86, 2008.

[51] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886. IEEE Computer Society, 2012.

[52] Raphaël Féraud, Olivier J. Bernier, Jean-Emmanuel Viallet, and Michel Collobert. A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1) :42–53, 2001.

[53] Enrique Sánchez-Lozano, Brais Martínez, Georgios Tzimiropoulos, and Michel F. Valstar. Cascaded continuous regression for real-time incremental face tracking. In *ECCV, Proceedings*, pages 645–661, 2016.

[54] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal. Robust online face tracking-by-detection. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2016.

[55] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn : Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99) :1–1, 2016.

[56] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In *IEEE International Conference on Computer Vision,*, pages 273–280 vol.1, Oct 2003.

[57] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible : Learning where the object might be. In *Computer Society Conference on Computer Vision and Pattern Recognition, IEEE*, pages 1285–1292, June 2010.

[58] Lu Zhang and L. van der Maaten. Structure preserving object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1838–1845, June 2013.

[59] T. B. Dinh, N. Vo, and G. Medioni. Context tracker : Exploring supporters and distracters in unconstrained environments. In *CVPR*, pages 1177–1184, June 2011.

[60] W. Bouachir and G. A. Bilodeau. Structure-aware keypoint tracking for partial occlusion handling. In *IEEE Winter Conference on Applications of Computer Vision*, pages 877–884, March 2014.

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

[62] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *Int. Journal of Computer Vision*, 77(1-3) :125–141, May 2008.

[63] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1838–1845, June 2012.

[64] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, Washington, DC, USA, 2014. IEEE Computer Society.

[65] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Proceedings of International Conference on Neural Information Processing Systems*, NIPS, pages 809–817, USA, 2013. Curran Associates Inc.

[66] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of European Conference on Computer Vision - Volume Part IV*, ECCV, pages 702–715, Berlin, Heidelberg, 2012. Springer-Verlag.

[67] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer. Adaptive color attributes for real-time visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, June 2014.

[68] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking : A benchmark. *IEEE Conference on Computer Vision and Pattern Recognition*, 0 :2411–2418, 2013.

[69] Mottaghi Roozbeh, Fidler Sanja, Yao Jian, Urtasun Raquel, and Devi Parikh. Analyzing semantic segmentation using hybrid human-machine crfs. *IEEE Conference on Conference on Computer Vision and Pattern Recognition, 2013*, pages 3143–3150.

[70] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7) :1195–1209, July 2009.

[71] X. Jia, H. Lu, and M. H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1822–1829, June 2012.

[72] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *Proceedings of the International Conference on Computer Vision*, ICCV, pages 1323–1330, Washington, DC, USA, 2011. IEEE Computer Society.

[73] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proceedings of the British Machine Vision Conference*, pages 6.1–6.10. BMVA Press, 2006.

[74] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *Proceedings of the European Conference on Computer Vision*, ECCV, pages 759–773, Berlin, Heidelberg, 2012. Springer-Verlag.

[75] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *CoRR*, abs/1501.04587, 2015.

[76] Horst Possegger, Thomas Mauthner, and Horst Bischof. In defense of color-based model-free tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2113–2120, June 2015.

[77] P.-L. St-Charles and G.-A. Bilodeau. Improving background subtraction using local binary similarity patterns. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 509–515, March 2014.

[78] Jianbo Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

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

[80] Georg Nebehay and Roman Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *IEEE Winter Conference on Applications of Computer Vision*, March 2014.

[81] Baiyang Liu, Junzhou Huang, Lin Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1313–1320, Washington, DC, USA, 2011. IEEE Computer Society.

[82] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking. In *Proceedings of the European Conference on Computer Vision*, ECCV, pages 864–877, Berlin, Heidelberg, 2012. Springer-Verlag.

[83] Zhaowei Cai, Longyin Wen, Jianwei Yang, Zhen Lei, and StanZ. Li. Structured visual tracking with dynamic graph. In *ACCV*, volume 7726 of *Lecture Notes in Computer Science*, pages 86–97. Springer Berlin Heidelberg, 2013.

[84] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *Proceedings of the European Conference on Computer Vision*, ECCV, pages 234–247, Berlin, Heidelberg, 2008. Springer-Verlag.

[85] Ju Hong Yoon, Du Yong Kim, and Kuk-Jin Yoon. Visual tracking via adaptive tracker selection with multiple features. In *ECCV (4)*, volume 7575 of *Lecture Notes in Computer Science*, pages 28–41. Springer, 2012.

[86] Junseok Kwon and Kyoung Mu Lee. Tracking by sampling trackers. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1195–1202, Nov 2011.

[87] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *Proceedings of the European Conference on Computer Vision*, ECCV, pages 661–675, London, UK, UK, 2002. Springer-Verlag.

[88] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5) :564–575, May 2003.

[89] S. Stalder, H. Grabner, and L. v. Gool. Beyond semi-supervised tracking : Tracking should be as simple as detection, but not simpler than recognition. In *IEEE Conference on Computer Vision Workshops, ICCV Workshops*, pages 1409–1416, Sept 2009.

[90] R. T. Collins. Mean-shift blob tracking through scale space. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–234–40 vol.2, June 2003.

[91] Y. Wu, B. Shen, and H. Ling. Online robust image alignment via iterative convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1814, June 2012.

[92] D. Wang, H. Lu, and M. H. Yang. Least soft-threshold squares tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2371–2378, June 2013.

[93] J. Lu, G. Wang, W. Deng, and K. Jia. Reconstruction-based metric learning for unconstrained face verification. *IEEE Transactions on Information Forensics and Security*, 10(1) :79–89, Jan 2015.

[94] E. Elhamifar and R. Vidal. Sparse subspace clustering : Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11) :2765–2781, Nov 2013.

[95] M. Ali Akber Dewan, E. Granger, G.-L. Marcialis, R. Sabourin, and F. Roli. Adaptive appearance model tracking for still-to-video face recognition. *Pattern Recognition*, 49 :129 – 151, 2016.

[96] Lawrence B. Holder and Diane J. Cook. Graph-based relational learning : Current and future directions. *SIGKDD Explor. Newsl.*, 5(1) :90–93, July 2003.

[97] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6) :1373–1396, June 2003.

[98] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV, Washington, DC, USA, 1999. IEEE Computer Society.