Victor Hugo Ayma Quirita

Collaborative Face Tracking: A Framework for the Long-Term Face Tracking

TESE DE DOUTORADO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
Programa de Pós-graduação em Engenharia Elétrica

Rio de Janeiro
December 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Victor Hugo Ayma Quirita

Collaborative Face Tracking: A Framework for the Long-Term Face Tracking

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica.

Advisor : Prof. Raul Queiroz Feitosa
Co-advisor: Dr. Patrick Nigri Happ

Rio de Janeiro
December 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Victor Hugo Ayma Quirita

Collaborative Face Tracking: A framework for the long-term face tracking

Thesis presented to the Programa de Pós-Graduação em Engenharia Elétrica of PUC-Rio, in partial fulfillment of the requirements for the degree of Doutor em Engenharia Elétrica. Approved by the undersigned Examination Committee.

Prof. Raul Queiroz Feitosa
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Dr. Patrick Nigri Happ
Co-Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Bruno Feijó
Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo
Departamento de Informática – PUC-Rio

Prof. Ricardo Farias
UFRJ

Prof. Gilson Alexandre Ostwald Pedro da Costa
UERJ

Prof. Márcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, December 18th, 2018.

Victor Hugo Ayma Quirita

The author received his bachelor's degree in Electronic Engineering at the Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) in 2011. He obtained his master's degree in Electrical Engineering with emphasis on Signal Processing and Control at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) in 2014. Since then, he has been working on the field of computer vision and pattern recognition.

Bibliographic data

Ayma Quirita, Victor Hugo

Collaborative Face Tracking: A Framework for the Long-Term Face Tracking / Victor Hugo Ayma Quirita; advisor: Raul Queiroz Feitosa; co-advisor: Patrick Nigri Happ. – Rio de janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2018.

v., 81 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Rastreamento de objetos;. 3. Rastreamento de Faces;. 4. Detecção de Faces;. 5. Fusão de Rastreadores. I. Feitosa, Raul Queiroz. II. Happ, Patrick Nigri. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 620.11

To God, for the gift of life, mystery, and curiosity.

To Victor and Rina, for teaching me unwrapping the present.

To Andrés, Pao, and Isa, for appreciating such a present along with me.

To Diego and Daniel, interestingly, life is mostly about detecting and tracking.

Acknowledgments

For whom I am, the blame is all on you: Victor and Rina. It would take more than a lifetime to recognize all your loving, effort, dedication and sacrifice in contributing to my personal and professional growth. For that, and for what that might trigger, I will always be grateful to you both. I'm about to become a Doctor. Yay!

I'm especially grateful to my advisor, Prof. Raul Queiroz Feitosa, for giving me the opportunity of learning and pursuing the most important goal in my academic career, which demanded his patience, support, understanding and encouraging words over these years of scientific research guidance.

My heartfelt thanks to my co-advisor, Ph.D. Patrick Nigri Happ, for his undivided attention, critical thinking, and productive scientific discussions during our meetings, which considerably contributed to the consequence of this doctoral research.

I cannot forget to extend my gratitude to Prof. Gilson Alexandre Ostwald Pedro da Costa, for his scientific perspective, advice, enthusiasm and encouraging words from the very beginning of this research.

There are few things in life as important as family, whose roots transcend frontiers. I am deeply grateful to my siblings: Andres, Paola, and Isarina, whose constant love and support were essential towards the conclusion of my research. I would also like to thank Gladys and Grover, for helping my parents watering and grow the plants in our early years. I offer my most sincere gratitude to Maybee, Gerald and my family by extension, for their positive thoughts and encouraging words in pursuing this goal.

I am grateful to my colleagues in the LVC, who have always been willing to listen and help, and with whom I have shared exceptional coffee afternoons, which most of the times were full of joy, anecdotes, and endless scientific discussions. I am especially thankful to Pedro Achanccaray and Jose Bermudez for their friendship during these years of personal and academic coexistence.

I only have words of appreciation for Walter and Martha, whose friendship have made my stay in Rio de Janeiro warmer, and for Vidali and Jossi, who have always been willing to share their positive thoughts so I could get back on track. Thank you, guys.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001*.

This work would not have been possible without support of Pontifical Catholic University of Rio de Janeiro (PUC-Rio).

Lastly, for all the experiences, learnings and friendships that I have accumulated over the last years here, I have only one left thing to say: "Obrigado, Rio de Janeiro, Cidade Maravilhosa."

Abstract

Ayma Quirita, Victor Hugo; Feitosa, Raul Queiroz (Advisor); Happ, Patrick Nigri (Co-Advisor). **Collaborative Face Tracking: A Framework for the Long-Term Face Tracking**. Rio de Janeiro, 2018. 81p. Tese de doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Visual tracking is fundamental in several computer vision applications. In particular, face tracking is challenging because of the variations in facial appearance, due to age, ethnicity, gender, facial hair, and cosmetics, as well as appearance variations in long video sequences caused by facial deformations, lighting conditions, abrupt movements, and occlusions. Generally, trackers are robust to some of these factors but do not achieve satisfactory results when dealing with combined occurrences. An alternative is to combine the results of different trackers to achieve more robust outcomes. This work fits into this context and proposes a new method for scalable, robust and accurate tracker fusion able to combine trackers regardless of their models. The method further provides the integration of face detectors into the fusion model to increase the tracking accuracy. The proposed method was implemented for validation purposes and was tested in different configurations that combined up to five different trackers and one face detector. In tests on four video sequences that present different imaging conditions the method outperformed the trackers used individually.

Keywords

Object Tracking; Face Tracking; Face Detection; Tracking Fusion

Resumo

Ayma Quirita, Victor Hugo; Feitosa, Raul Queiroz; Happ, Patrick Nigri. **Rastreamento de Faces Colaborativo: Uma Metodologia para o Rastreamento de Faces ao Longo Prazo**. Rio de Janeiro, 2018. 81p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

O rastreamento visual é uma etapa essencial em diversas aplicações de visão computacional. Em particular, o rastreamento facial é considerado uma tarefa desafiadora devido às variações na aparência da face, devidas à etnia, gênero, presença de bigode ou barba e cosméticos, além de variações na aparência ao longo da sequência de vídeo, como deformações, variações em iluminação, movimentos abruptos e oclusões. Geralmente, os rastreadores são robustos a alguns destes fatores, porém não alcançam resultados satisfatórios ao lidar com múltiplos fatores ao mesmo tempo. Uma alternativa é combinar as respostas de diferentes rastreadores para alcançar resultados mais robustos. Este trabalho se insere neste contexto e propõe um novo método para a fusão de rastreadores escalável, robusto, preciso e capaz de manipular rastreadores independentemente de seus modelos. O método prevê ainda a integração de detectores de faces ao modelo de fusão de forma a aumentar a acurácia do rastreamento. O método proposto foi implementado para fins de validação, tendo sido testado em diversas configurações que combinaram até cinco rastreadores distintos e um detector de faces. Em testes realizados a partir de quatro sequências de vídeo que apresentam condições diversas de imageamento o método superou em acurácia os rastreadores utilizados individualmente.

Palavras-chave

Rastreamento de objetos; Rastreamento de Faces; Detecção de Faces; Fusão de Rastreadores

Table of contents

List of figures

List of tables

1
Introduction

Social organization has evolved over time. After nomads mastered farming and domestication, they changed their wandering nature for living in settlements. The establishment of the first settlements led to the organization and control of people interactions to ensure the proper use of shared resources.

As the settlements began to expand, there was a need for protection mechanisms against harm from internal or external forces. Thus, guards and sentinels were usually employed to watch and secure people and their environment. With the passage of time, drawings of known criminals began to be distributed, so that they could be recognized and hunted. Then, as the technology advanced, automatic and semi-automatic solutions became available to aid and alleviate the human efforts.

Nowadays, the explosion of population growth has complicated the control of human interactions. On the other hand, the evolution of knowledge allowed the development of new technologies for information management, which enables some level of control of social interactions and provides greater security in political, monetary, technological and physical infrastructure contexts.

In this sense, computer vision scientists have been working for decades on the development of methods that enable machines to understand the world through the analysis of a single or paired still images (Jain et al., 1995). However, the lack of temporal context for describing the complex world dynamics limits their application, exposing the need to work with image sequences, i.e., videos. Techniques based on video sequences are capable of detecting and tracking changes associated with objects of particular interest. They allow extracting meaningful information which are often used by law enforcement applications, among others.

Over the past couple of decades there has been a rising interest in collecting and processing data associated with human faces, driven mostly by national and international security, law enforcement and facial authentication issues. Moreover, face detection and recognition have proven to be valuable, reliable and flexible biometric applications, based on image data, which is easy to acquire in a non intrusive way (Li and Jain, 2005).

Over the years, face recognition has become a relatively well-solved

problem, with a series of algorithms achieving a very high degree of accuracy (Taigman et al., 2014; He et al., 2016; Liu et al., 2017; Ranjan et al., 2017; Wang and Deng, 2018). Similarly, face detectors have been exhaustively studied in the literature, achieving remarkable performance in locating faces within an image (Zhang and Zhang, 2010; Zafeiriou et al., 2015). However, they lack associative capabilities to assign and hold an identifier to a face of interest over time. In order to solve this data association requirement, face trackers are designed to estimate the state of a face of interest within a sequence of images. The state might contain properties of the face, such as position, extent, velocity, appearance, orientation, among others (Forsyth and Ponce, 2011), which helps locating and following the target face over time.

In fact, face tracking is a particular case of visual object tracking, also known as object tracking. This is the reason why the literature refers to a face as being an object within the object tracking context. Typically, the face tracking process starts with just one sample of a facial image in a given frame of the video sequence. Then, a tracker creates a model of the target face and incrementally updates it with the face variations perceived over the frames. In this respect, a tracker should be able to correctly track a face indefinitely as long as it remains visible. The literature attributes the name of long-term tracking to this task (Kalal et al., 2012).

Although tremendous efforts have been made to produce robust and accurate tracking algorithms, long-term tracking in real-world scenarios remains challenging due to intrinsic and extrinsic factors that influence the appearance of the object, consequently leading the trackers to drift away from the target or to the loss of its track (Smeulders et al., 2014).

Intrinsic factors correspond to deformations caused by rigid (translation and rotation) and non-rigid transformations and to inherent object properties such as age, ethnicity, gender and facial hair, in the case of the face tracking. External factors, which are often uncontrolled in real-world scenarios, correspond to the movement of the camera during acquisition, to temporal and spatial image resolutions, to image noise due to sensor's characteristics, to loss of information caused by three-dimensional space projection onto a bi-dimensional image, to changes in illumination, to similarity in appearance and to occlusions (Jain et al., 1995; Yilmaz et al., 2006; Magio and Cavallaro, 2011).

Comparative studies have shown that different trackers exhibit different behaviors when facing some of the aforementioned problems, indicating that a single tracker may not cope with all kinds of perturbations that occur during the tracking process and generally master just one or few of them (Smeulders et al., 2014; Wu et al., 2015; Li et al., 2016; Kristan et al., 2016). Additionally,

their results also suggest that there is a complementary behavior among trackers, which might be exploited to increase overall tracking performance.

In this sense, several authors have proposed to exploit this complementary behavior by putting different trackers to work together in an ensemble. Some authors have combined trackers' estimates into a single and, expectedly, more reliable estimate (Shearer et al., 2001; Leichter et al., 2006; Stenger et al., 2009; Li et al., 2012; Bailer et al., 2014; Gao et al., 2014). To further improve the overall tracking performance, other authors have proposed to enhance individual tracker's performance by resubmitting the combined estimate back to the trackers in the ensemble (Zhong et al., 2014; Biresaw et al., 2015; Leang et al., 2015; Ayma et al., 2017). Somewhat different from the combination of tracking estimates, few authors have incorporated prior knowledge of the object, such as the inclusion of detectors, within the tracking process to correct tracker's trajectory (Kalal et al., 2010; Fan and Ling, 2017). Although a lot of work have been made in the recent years, the long-term tracking and specially the long-term face tracking is still an open problem.

In this work, we combine some of these ideas to propose a novel and robust framework for the long-term face tracking in unconstrained scenarios, specially scenarios that contain occluding objects and multiple-faces. The main idea is to use the information provided by an offline face detector to complement the trackers conforming an ensemble. This approach would allow each tracker to adjust its tracking trajectory, recover from tracking failures or recapture the track of the target face after short periods of disappearance, improving, in this way, individual trackers and consequently the unified tracking response. In a nutshell, the proposed framework comprises four modules that operate over each frame of a video sequence. These modules are responsible for: independently processing multiple tracking tasks of a particular face; merging the different trackers' estimates; running a face detection task in attempt to locate the target face; and finally, combining the detector's outcome with the merged tracking estimate to produce the final estimate and then to feed it back to each tracker in the ensemble.

1.1 Objectives

This thesis research aims to propose a framework for the long-term tracking of faces that provides accuracy and robustness in unconstrained scenarios, specially those scenarios that contain occluding objects and multiple-faces, by merging the outcomes from an ensemble of trackers and complementing it with information delivered by an offline face detector.

In pursuit of the general objective, the specific objectives are:

1. Conceive and develop an operational structure to allow trackers to correct their state, update their facial appearance model, and re-initialize themselves when necessary.

2. Design a collaborative scheme to allow individual execution of trackers, a fusion of trackers' estimates, the execution of a face detector, and the combination between the face detector's and fusion's outcomes.

3. Conceive a verification mechanism to select the outcome from a pool of detection outcomes that corresponds to the target face.

4. Design a feedback mechanism to update or re-initialize the trackers individually based on the framework's output.

5. Build a prototype in C++ that implements the proposed framework and allows the scalability regarding the addition of different new trackers and the replacement of the face detector and fusion methods.

6. Build a facial video dataset collection by selecting public single-face and multiple-face video sequences and creating the reference annotations when they are not available.

7. Evaluating the proposed framework and comparing its performance with the individual trackers and the tracking fusion method adopted.

1.2 Thesis Contributions

The main contributions of this work are fourfold:

1. Propose, implement and evaluate a novel framework for the long-term face tracking, which comprises an ensemble of generic object trackers and face detectors.

2. Devise an association and verification mechanism that allows selecting the target face among a set of candidates provided by a face detection algorithm that enables the long-term face tracking.

3. Investigate decision-making alternatives to update or reset the trackers that compose the ensemble based on the framework's combined outcome.

4. Make available a face-specific tracking database, assembled by selecting facial video sequences available in the literature and annotating their references when they are not available.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. The next chapter makes a review of the literature presenting the most relevant works related to the face tracking problem, including the state-of-the-art on object detection and tracking, as well as tracker fusion and combination of face trackers with face detectors. Chapter 3 describes the proposed framework for the long-term face tracking and its modules. Chapter 4 presents the experimental analysis carried out to evaluate the proposed framework. Finally, Chapter 5 summarizes the conclusions drawn from the experimental results and give directions for future works.

2
Related Works

Face tracking is fundamental for numerous applications in fields as diverse as security, health, sports, digital gaming, marketing, and so on. For example, face verification systems use face tracking methods to collect facial data in order to validate the identity claimed by a person. In video surveillance systems, face tracking algorithms provide facial data to analyze facial expressions and, consequently, recognize suspicious behaviors.

Although face tracking, as its name suggests, works over the facial images domain, it is a subfield of the visual object tracking; hence, it inherits all of its properties and methods for the tracking task. Furthermore, this allows regarding a face [1] as an object within the visual object tracking.

The vast quantity of published works on visual object tracking such as books (Ballard and Brown, 1982; Magio and Cavallaro, 2011; Forsyth and Ponce, 2011) surveys (Yilmaz et al., 2006; Yang et al., 2011; Smeulders et al., 2014; Wu et al., 2015; Li et al., 2016; Kristan et al., 2016; Li et al., 2018) journal papers, and conferences show that visual object tracking is a field of great interest and remains in continuous development.

This chapter provides an overview of visual object tracking literature, including related works that are considered the state-of-the-art in face detection, fusion of tracking algorithms and the collaboration between tracking and detection algorithms.

2.1
Face Detection

Face detection refers to the process of locating the regions within an image that encompass facial patterns. Typical face detection algorithms are trained to learn facial patterns from a representative training set, which is expected to gather most of the faces' variability in unconstrained scenarios (Zafeiriou et al., 2015; Zhang and Zhang, 2010; Yang et al., 2002).

Conventional face detection algorithms submit a set of image patches to a binary classifier, which outcome tells whether a patch corresponds to a face or

[1] In this work, we acknowledge that a face is also an object in its most abstract form and as such it uses the words face and object indistinctly.

not. The set of image patches can be extracted using a sliding window which is moved through the image at fixed position steps and scales (Forsyth and Ponce, 2011). For example, Viola and Jones (2001) combined several weaker binary classifiers into a cascade structure to discriminate if an image patch corresponds to a face. Although the algorithm performs well in constrained scenarios, it fails in the presence of faces with large angle variations, partial occlusions and appearance changes caused by lighting conditions.

Aiming at improving Viola and Jones's face detector, Li and co-authors (2011) combined Speeded up Robust Features (SURF) with a cascade of weak classifiers. Jun et al. (2013) proposed variants of the Local Binary Pattern (LBP) and Gradient of Histograms (HOG) features, namely Local Gradient Patterns (LGP) and Binary Histograms of Oriented Gradients (BHOG), and a hybrid feature that combines the LBP, LGP, and BHOG via the AdaBoost algorithm. Despite the efforts to produce robust face detectors, the aforementioned methods still have problems when facing significant facial pose variations, deformations, and occlusions.

Recent studies in areas related to face detection highlight the discriminative power of cascade classifiers, as well as the trade-off between the number of the cascade stages and the quality of the features used to capture the variations of the objects (Zafeiriou et al., 2015; Zhang and Zhang, 2010). In particular, Convolutional Neural Networks (CNN) have shown a remarkable performance in object detection tasks, mostly attributed to their capacity to learn objects' representations. Li et al. (2015) introduced a CNN-based cascade classifier to locate faces in an image. The classifier comprises three stages, containing two CNN's each. The first CNN in each stage refines candidate facial regions as they pass through the cascade, whereas the second CNN performs a bounding box correction, also known as bounding box calibration, for a better alignment with the actual face in the image.

To improve the CNN-based face detectors' performances, several authors have proposed to learn correlated features with face detection in a simultaneous manner, such as facial landmark location, head posture estimation, gender recognition, among others. For example, the DNN face proposed by Zhang et al. (2016) exploits the inherent correlation between face detection and facial landmarks location via a deep cascade multitask framework. Analogous to the method presented by Li et al. (2015), Zhang's CNN architecture comprises three stages, which refines candidate facial regions and calibrate their corresponding bounding boxes. However, the final stage in Zhang's approach performs a landmark localization to improve its discriminative power. In a similar manner, Ranjan et al. (2017) designed a CNN architecture to

detect faces while locating facial landmarks, estimating the head posture and recognizing the gender at the same time. The authors proposed to learn common features for these tasks by fusing the feature maps throughout the network using a separate fusion-CNN.

In short, face detectors have increasingly improved over time. The emergence of deep learning based technologies and the availability of training data covering a broader range of facial appearances has allowed the creation of robust face detectors. Given its capacity to locate faces, modern face detectors might operate over the frames of a video sequence to collect facial data, however they lack mechanisms to associate the target face to a correspondent detection. In this sense, tracking algorithms are required since they are specifically designed to collect facial data and to track the changes related to a face of interest.

2.2 Visual Object Tracking

In contrast to face detection, face tracking aims to estimate the state of a face as new frames become available. This implies solving a data association problem inherent to the tracking process: a tracker must ensure that a face stays associated with a unique identifier along the frames of a video sequence.

Visual object tracking, or simply tracking, caught the community's attention with the seminal work of Lucas and Kanade on image registration using matching techniques in the early 80's (1981). The general idea was to minimize the mismatch between a reference and a candidate template. The major drawback of this and similar approaches are related to the use of a single reference template, which aims to capture the appearance variations of the object during tracking (Comaniciu et al., 2000; Baker and Matthews, 2004; Matthews et al., 2004). Furthermore, the object is often described by color histograms or image patches, which are sensitive to illumination changes, occlusions, abrupt motion, and changes in object's size.

Nowadays, a typical tracker creates an appearance model of the target object based on the information extracted from a bounding box given at the beginning of the tracking process. This model is incrementally updated with data available during the tracking. Although the appearance model is specific to a tracker, it often gets affected by object's deformations and fast movements, occlusions and lighting variations, as well as image resolution and sensor noise, among others factors. So, the success of a tracking algorithm depends on its capacity to adapt to the changes in the object's appearance that occur during tracking.

Several authors have explored the idea of using online techniques from the machine learning field to learn the changes in the object's appearance. The literature categorizes such methods as discriminative and generative trackers. While discriminative trackers train an online binary classifier to distinguish the object from the background, generative trackers model the object appearance during tracking disregarding background's information. In both cases, an incremental update with reliable data is crucial in order to prevent tracking failures (Yang et al., 2011).

Discriminative trackers consider the task of tracking as a binary classification problem. The estimate of the object's state, often represented by a bounding box, corresponds to the location of the image patch that gets the maximum classification score within a local neighborhood near to the previous state. It is important to note that discriminative trackers strongly depend on designing features robust enough to represent the object's appearance, as well as the mining of reliable data for online classifier training. For example, Collins et al. (2005) and Grabner et al. (2006) proposed methods to select the best-suited features in an online manner. In addition to feature selection, Babenko et al. (2011) explored multiple instance learning algorithms to resolve the uncertainty in self-generated training data. Moreover, Kalal et al. (2012) proposed to combine a tracker and an online detector's responses to improve tracking performance, however both the tracker and the detector are subjected to appearance variations present during tracking. Following this line, Zhang and co-authors (2014) proposed to model the object's appearance in a compressed domain, resulting from the projection of image features to a randomly chosen low-dimensional space. Hare et al. (2016) integrated the learning and tracking process by using structured classification outcomes avoiding to generate labeled data for training.

Generative trackers, on the other hand, aim at modeling the object's appearance in a d-dimensional space using data available only during tracking. Conventional generative tracking algorithms measure the similarity between the model and a candidate image patch to estimate the state of the object where the similarity measure is often a pre-defined matching function. For example, Ross et al. (2008) designed a method that incrementally captures the appearance variations by learning a low-dimensional subspace representation. Mei and Ling (2009) estimate the object's state by finding the lowest projection error between a candidate template and target templates on a sparse space, spanned by the target and trivial templates. Henriques et al. (2015) employed non-linear mapping and trained an online linear ridge regression model to predict the next state of the object. Recently, He and co-authors (2017) pre-

sented a novel approach to improve tracking performance under illumination changes and occlusions. The method analyzes a multi-region representation, called local sensitive histograms, which relates pixels intensities and positions.

Although the use of hand-crafted features has led to considerable performance improvements in the task of both discriminative and generative tracking (Smeulders et al., 2014; Wu et al., 2015), the adoption of cutting-edge methods for representation learning within the tracking process has presented promising results (Li et al., 2018). Recently, CNNs have proven to be powerful tools to learn features calling the attention of the visual object tracking community. For instance, Ma and co-authors (2015) presented a coarse-to-fine approach for object tracking that combines the responses of a set of online learned correlation filters which correspond to the 3rd, 4th, and 5th convolutional layers of the VGG-Net (Simonyan and Zisserman, 2015). Their approach exploits the CNN's capability to encode relevant object's information along the convolutional layers of the network. The last convolutional layers of a CNN encode the semantic information of the object being robust to significant changes in appearance, while early layers capture spatial details which are suitable for a precise object location.

Following the deep learning approach, Danelljan et al. (2016) proposed to fuse the outputs of multiple convolutional layers into a joint learning framework instead of dealing with separate correlation filters. This scheme integrates multi-resolution deep feature maps into a learning process of continuous convolution filters on the spatial domain, which produces a continuous-domain confidence map of the object's state.

In the recent past, some authors have extended the use of the CNN's within a siamese architecture to learn a matching function for tracking. The matching function measures the similarity between a reference image of the target object and a candidate image extracted from a new frame of a video sequence. In this sense, the function provides a high score if the two images correspond to the same object, or produces a low score, otherwise. For example, Tao and co-authors (2016) employed a siamese network to learn a matching function that returns the location of the most similar image candidate with the reference. In this approach, they used data from several video sequences to train the siamese network in an offline manner. Bertinetto et al. (2016) proposed a similar approach using a fully convolutional network for a dense search of the best image candidate.

As presented by the aforementioned methods, conventional and deep learning based techniques perform well on the tracking of arbitrary objects for short periods. However, they often fail in the presence of severe perturba-

tions, such as occlusions, object's transformations and image resolution, which restricts them to operate on the long-term. The comparative studies of (Wu et al., 2015; Smeulders et al., 2014; Kristan et al., 2016; Li et al., 2016) have exposed the limitations of individual trackers to cope with these changes in an object's appearance showing that the occurrence of simultaneous perturbations might decrease the tracking performance considerably. Moreover, their results also suggest that different trackers have complementary behaviors, as some trackers perform well in situations where others perform poorly. Thus, the fusion of complementary trackers would improve significantly the accuracy and robustness of the tracking process, specially under the effect of different perturbations.

2.3 Tracking Fusion

Following the premise that a tracker ensemble would result in a more robust performance than single trackers, several authors have proposed fusion techniques over the years. Shearer et al. (2001) proposed to select one of the estimates between a region-based tracker and an edge-based tracker according to a confidence measure. However, it requires user intervention when possible drifts are detected. Leichter et al. (2006) combined several tracking estimates through the exchange of their final state *pdf* (probability density function). The method is, however, limited to trackers of the same nature.

On a more general fusion framework, Stenger and co-authors (2009) learned the error distributions of a collection of trackers from a representative training set in order to select the best-suited trackers for a given application. Nevertheless, the proposed approach is limited to the range of perturbations present during training, and to a certain number of trackers. Similar to Stenger and co-authors, Li et al. (2012) proposed a disagreement-based fusion approach, which has also restrictions in terms of the number and type of trackers. Bailer and co-authors (2014) combined the estimates of a set of trackers through an offline trajectory optimization scheme, where tracking results for a given video sequence were already known in advance.

The aforementioned methods attempt to fuse a set of tracking estimates into a single and more reliable estimate. They are, however, limited to particular tracking designs or to specific offline training procedures. To overcome these problems, Gao et al. (2014) proposed a method to fuse the estimates of an ensemble of trackers, which is independent of their particular natures, i.e., trackers are treated as black boxes.

In short, the fusion of visual object tracking algorithms aims to improve

the overall tracking accuracy by merging different tracker outputs. Although these methods usually improve the accuracy of the final outcome, each tracker is still subjected to failures often caused by error accumulation in its object's model, which makes its tracking estimates to drift away from the target. In fact, the long-term tracking requires recovering from tracking failure or reacquiring the target once it reappears in the camera's field of view. (Li et al., 2018; Wu et al., 2015; Smeulders et al., 2014; Kalal et al., 2012). Such problems can be mitigated by feeding the tracker from time to time with reliable training samples that might be provided by a unified response of an ensemble of trackers, an object-specific detector or a combination of both. In the next section, some methods based on this feedback process are presented.

2.4 Feedback Learning

Although the aforementioned Gao's approach (2014) is fairly general, it does not support updating the object's representation of the individual trackers, which can lead them to drift away from the target. In this direction, some works include a feedback mechanism, based on the fusion output, to provide more reliable training samples to eventually correct each tracker. Thus, this technique enhances the individual tracker performance and, consequently, the final result.

In this sense, Leang et al. (2015) evaluated different strategies for updating or re-initializing trackers by combining fusion outputs and drift predictors. Each tracker's contribution is given by a binary confidence level, which considers individual trackers' performance in the previous and current frames instead of the accumulated performance during tracking.

In the same line, Zhong and co-authors (2014) proposed a probabilistic approach for the fusion of the trackers' estimates. In their approach, the fused tracking estimate is also used to update the trackers in the ensemble in order to improve their accuracy. Biresaw et al. (2015) proposed a fusion framework that enables individual tracker correction based on estimates provided by other trackers, but the method is restricted to Bayesian trackers. Finally, Ayma and co-authors (2017) have extended Gao's fusion approach by resubmitting the fusion's estimate to the trackers that comprise an ensemble. A problem with this approach is that, depending on the ensemble design, trackers with poor performances may affect the results as much as well-behaved trackers, leading to sub-optimal outcomes.

2.5

Combining external Detectors with Trackers

The idea of combining visual object tracking and object detection algorithms has been also explored in the literature. However, the inability of the detectors to solve the data association problem has lead scientists to focus on developing methods to ensure a correspondence between the target object with only one of the detection outcomes.

Recently, Fan and Ling (Fan and Ling, 2017) presented a framework that combines a fast operating tracker with a tracker's estimate verifier to perform the long-term object tracking. In this approach, the verifier is executed periodically and in parallel with the tracker. Moreover, the verifier examines the tracker's estimates aiming for accuracy. Thus, in the case of a tracking failure, the verifier adopts a detector behavior to provide the tracker with an alternative tracking estimate to correct its state and continue with the tracking.

All the methods presented above in this chapter were developed to work with arbitrary objects. In this sense, an object represents anything that is of interest for a given application, such as products, cars, persons, among others. Therefore, these tracking algorithms could be adapted to face tracking. However, there are few works that focus on face tracking applications in the literature. Some of them are presented in the next section.

2.6 Face Tracking

Combining some of the already presented ideas, Kalal et al. (Kalal et al., 2010) proposed an scheme to track faces in a video sequence, using an offline face detector and focusing on the development of a validator, which certifies that the detector and the tracker outcomes correspond to the face of interest.

However, human faces are hard to control in real-world scenarios given its highly deformable nature, which complicate the tracking process. To overcome this inconvenience, some authors have wrapped around the face tracking task into a continuous face landmark localization problem. The landmarks correspond to fiducial face features, such as eyes, nose, mouth and so forth. Usually, face landmark localization algorithms are based on deformable models, which dates back to the work of Cootes et al., (2001). The deformable models can be learned offline to fit a facial image (Xiong and De la Torre, 2013) or can be incrementally updated to cope with face variations (Asthana et al., 2014). Nevertheless, both approaches often need the intervention of a face detector to produce reliable face tracking (Chrysos et al., 2018).

In this work, we want to improve the face tracking solutions by proposing a new framework focused on long-term face tracking, called Collaborative

Face Tracking. This framework combines some of the most prominent ideas presented in this chapter such as the fusion of different trackers, the feedback learning and the combination with a face detector. As far as we know, the proposed framework is the first of its kind presenting all these characteristics together to this aim. Additionally, the framework is flexible and scalable, since any tracking algorithm can be included in the framework, as well as any tracking fusion technique that accepts tracking estimates as input.

3
Collaborative Face Tracking

In the long-term face tracking, trackers should continuously deliver estimates about the state of the target face for long periods. However, face tracking algorithms are sensitive to different kinds of variations that frequently cause tracking failures such as changes in scene's illumination, occlusions caused by non-interest objects, resemblance among faces and deformations proper to the face dynamical behavior.

In fact, tracking failures are a consequence of error accumulation in the tracker's appearance model as a result of the model's update with inaccurate data, usually derived from the aforementioned variations. This often lead the trackers to lose or drift away from the target face, impairing in this way the long-term face tracking. In order to avoid those errors, some trackers explore different ways to constantly adjust their states estimates, update their facial models or combine both alternatives.

In this work, we propose a framework architecture to enable the long-term face tracking in unconstrained scenarios, including those where occlusions and multiple-faces are present. Our framework relies on the assumption that a consensus tracking estimate is more accurate and robust than individual trackers' estimates and can be used either to update or to re-initialize the trackers in the ensemble. Furthermore, complementary information about the face, provided by an offline face detector, may be used to refine the fusion process, recover from tracking failure and recapture the target face after a short disappearance.

The next sections in this chapter give an outlook of the framework architecture and present its components in details, as well as the methods that enable the trackers' update and re-initialization.

3.1
Framework Overview

The Collaborative Face Tracking is a framework for the long-term tracking of faces that takes into consideration the outcomes from an ensemble of trackers and from an offline face detector to produce a final tracking estimate, which is supposed to be more accurate and robust than individual trackers' es-

timates. The framework comprises four modules: Tracking, Fusion, Inspection, and Integration, organized in such a way that each tracker in the ensemble benefits from the final tracking estimate (see Fig. 3.1). The trackers are incrementally updated or re-initiated, depending on their agreement level with the final tracking estimate.

The tracking process starts with the Tracking module, which is composed of different trackers that will cooperate in an ensemble, initializing each tracker with information about the initial state (position and extent) of the target face in the first frame.

Next, for every incoming frame, a new estimation cycle is started. In the first step, the tracking module executes each tracker in the ensemble to produce a set of tracking estimates. These estimates are, in turn, processed by the Fusion module, which produces a consensus estimate of the possible state of the face (fusion estimate).

Simultaneously, the Inspection module, which is composed by an offline face detector, analyzes the current frame, upon request, to provide an additional, and expectedly reliable, guess (inspection estimate) about the current state of the face.

In the following, the Integration module combines the responses of both the Inspection and Fusion modules to generate the final estimate of the face state (integration estimate), which corresponds to the framework's output. Finally, the estimation cycle finishes once the Integration module forwards the final tracking estimate back to the Tracking module, which uses it to command each tracker in the ensemble to either update their models or re-initiate themselves.

3.2 Tracking Module

This module is in charge of managing the trackers conforming the ensemble, so they correctly operate within the framework. Most of the available trackers have a similar operating structure: first, they estimate the state of the target face for a given video frame, then they use this new estimate to update their facial models, as well as to produce a state estimate of the target face in the subsequent frame (refer to Figure 3.2(a)). However, this operating structure might contribute to the creation of unstable trackers due to the insertion of unreliable information into their facial models' from the use of their very own tracking estimates.

In order to ensure the correct framework's execution, we propose to extend the aforementioned operational structure by including a state correction

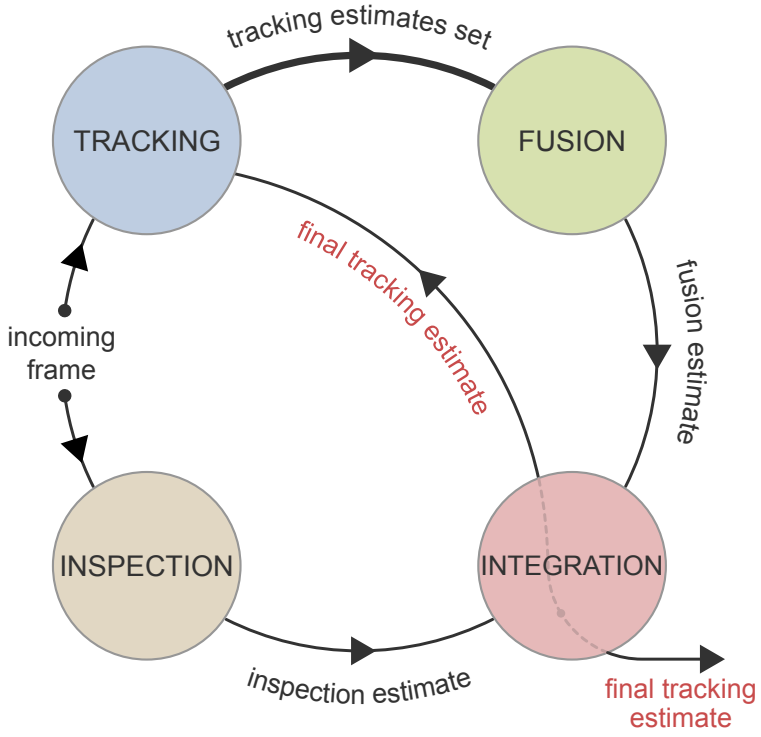


Figure 3.1: Collaborative Face Tracking, a framework for the long-term tracking of faces.

stage in between the state estimation and the model updating stages, as illustrated in Figure 3.2(b). In this way, the trackers benefit from reliable information delivered by the Integration module. Specifically, given a tracker in the ensemble, it first estimates the state of a target face in a given video frame and, at the end of the estimation cycle, it uses the outcome from the Integration module to correct the state of the target face (tracker's correction) and consequently update its facial model (tracker's update). Moreover, the tracker uses the corrected estimation as input in the estimation stage to produce the subsequent tracking outcome.

This new processing chain enables two ways for tracker's correction. In the first approach, a tracker corrects its tracking estimate in position-only, i.e., partial correction; whereas, in the second approach, the tracker corrects its position and extent, i.e., complete correction. The partial correction is defined by the translation of the tracking estimate towards the final tracking estimate provided by the Integration module, such that the center positions of the two estimates correspond with each other. The complete correction, on the other hand, involves replacing the tracker's state estimate with the position and extent of the integration estimate.

Although the outcome of the Integration module usually leads to a more accurate estimate of the target face state, the bare updating of a tracker might not be enough to bring it back to the correct track. In these scenarios, the

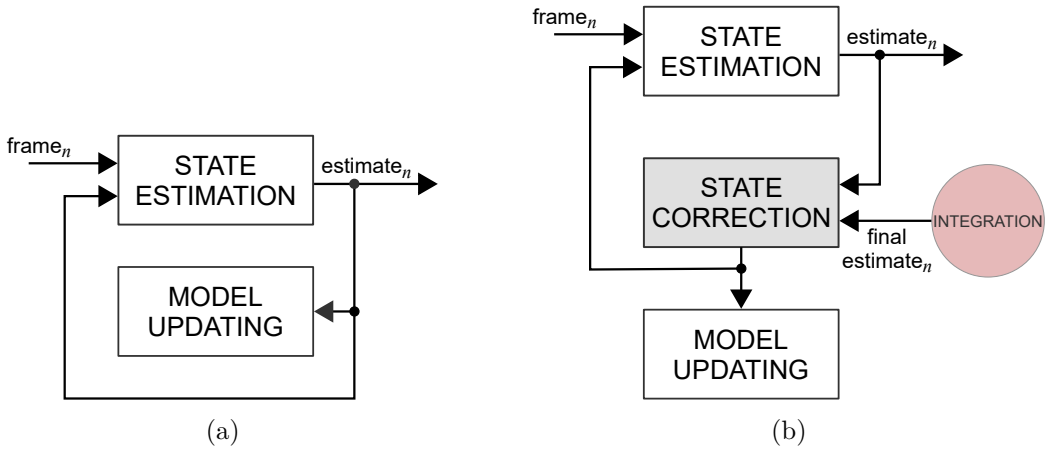


Figure 3.2: Trackers' operational structure. (a) Conventional operational structure. (b) Proposed operational structure.

tracker gets reinitiated. This process involves erasing the facial model and all of the accumulated records about the target face states and then starting over from scratch using the last estimate provided by the Integrator module as the new initial state. From now on, we refer to this process as tracker's re-initialization.

The decision between updating or re-initializing a tracker is subjected to an evaluation of its accumulated records about the state of the target face over some consecutive frames. Formally, given a tracker in the ensemble, first, the Tracking module measures the dissimilarity between an individual tracking estimate and the final tracking estimate forwarded by the Integration module. Next, the Tracking module stores the dissimilarity result into a cyclic buffer of fixed size (s_b) developed for the tracker, hereafter known as *cyclic dissimilarity buffer*. Then, the module computes a predefined statistic (mean, median, etc.) considering all the elements in the buffer and compares it against a predefined re-initialization threshold (δ_r). Finally, the Tracking module commands the tracker to update its facial model if the computed statistic is below δ_r; otherwise, the tracker gets re-initiated.

It is worth noting that the Tracking module is scalable regarding the number of trackers in the ensemble and flexible concerning their designs. Thus, it is possible to include as many different trackers as desired.

3.3 Fusion Module

The fundamental idea behind the Fusion module, and actually behind the whole framework, is that the cooperation among trackers with different characteristics and capabilities would lead to a better estimate of the real

state of the target face. So, the Fusion module takes the trackers' estimates as input and processes them together to generate the ensemble's single outcome, i.e., the fusion estimate.

In theory, this module is flexible in a way that any fusion method may be applied to produce a single result from the collective of trackers. In fact, the Fusion module solely demands the trackers' estimates and does not require any knowledge about the tracking algorithms in the ensemble, contributing in this way to the framework's scalability and flexibility.

In this work, we employed the fusion method proposed by Gao et al. Gao et al. (2014), namely the Symbiotic Tracker Ensemble, to process and merge the trackers' estimates. We chose this approach because it exploits both temporal and spatial relationships among trackers' estimates to produce a unified outcome. Furthermore, it disregards the trackers' designs, considering each tracker as a black box.

The Symbiotic Tracker Ensemble ponders the contribution of each tracker in the ensemble to the fusion estimate by weighting the trackers according to individual and collective evaluations of their behaviors, followed by a combination stage.

The individual evaluation, which corresponds to the *Intra-Tracker Correlation* stage in Figure 3.3, analyzes the tracker's consistency over time to provide an initial weight for each tracker, i.e., *Initial credibility*. To this end, the *Intra-Tracker Correlation* evaluates the smoothness in the tracker's trajectory by relating its estimates from consecutive frames with its previous *final credibility*, which is the weight computed by the *Inter-Tracker Correlation* stage.

The collective evaluation, which corresponds to the *Inter-Tracker Correlation* in Figure 3.3, measures the spatial congruence among the different trackers' estimates through multiple pair-wise trackers' interactions. It takes the initial trackers' credibilities into account in order to produce the *Final credibilities*, which correspond to the final trackers' weights.

In the sequence, the *Combination* stage presented in Figure 3.3 computes the *Fusion estimate* considering all the trackers' estimates and their respective *Final credibilities*.

Next, we describe a set of relationships between tracking estimates used in both *Intra-Tracker Correlation* and *Inter-Tracker Correlation* stages, followed by a brief explanation of how to compute both correlations and the fusion estimate.

3.3.1

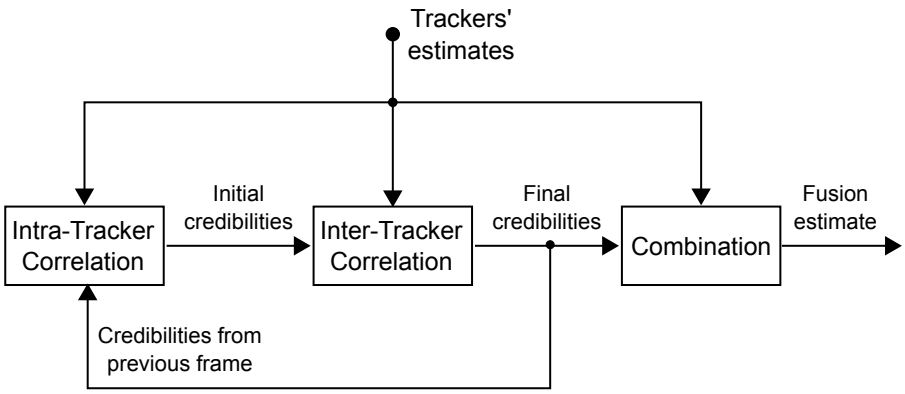


Figure 3.3: Symbiotic Tracker Ensemble.

Relationships Between Tracking Estimates

An important part of the Gao's fusion approach is to quantify relationships between two tracking estimates, R_1 and R_2, which can be represented by two bounding boxes in the form $R = (x, y, width, height)$.

For this purpose, Gao and coauthors introduce two similarity metrics, formally:

– $F(R_1, R_2)$: which measures the similarity between R_1 and R_2 as:

$$F(R_1, R_2) = \frac{2 \times Pr(R_1, R_2) \times Re(R_1, R_2)}{Pr(R_1, R_2) + Re(R_1, R_2)} \tag{3-1}$$

where $F(R_1, R_2) \in [0, 1]$, and $Pr(R_1, R_2)$ and $Re(R_1, R_2)$ represent precision and recall, respectively.

– $r(R_1, R_2)$: which quantifies the congruence between R_1 and R_2 according to:

$$r(R_1, R_2) = exp(-\frac{D^2(R_1, R_2)}{\sigma^2}) \tag{3-2}$$

where $r(R_1, R_2) \in [0, 1]$, $D(R_1, R_2)$ represents the Euclidean distance between the centers of R_1 and R_2, and σ is a controlling coefficient of the width of the exponential function.

3.3.2 Intra-Tracker Correlation

The first stage of Gao's fusion approach evaluates a tracker's consistency by assessing the changes in its trajectory. To this end, successive tracking estimates are used to compute a temporal correlation measure, which defines its initial credibility.

In a more formal way, given two consecutive tracking estimates $R_{i,n-1}$ and $R_{i,n}$, corresponding to the $i\text{-}th$ tracker in the ensemble at $(n-1)\text{-}th$ and $n\text{-}th$ frames, respectively, the initial credibility $C_{i,n}$ is defined by:

$$C_{i,n} = \xi_i \zeta_i + (1 - \xi_i)\Theta(R_{i,n-1}, R_{i,n})C^f_{i,n-1} \tag{3-3}$$

where, ζ_i is a tracker's prior credibility defined by the user, $C^f_{i,n-1}$ represents the final credibility coefficient from a previous frame, $\xi_i \in [0, 1]$ is a regularization parameter that controls the participation of the tracker's prior credibility, and $\Theta(\cdot)$ is a relation coefficient that assesses the trajectory smoothness of the *i-th* tracker.

The relation coefficient $\Theta(\cdot)$ can be computed using either $F(\cdot)$ or $r(\cdot)$ similarity metrics, as presented earlier in the previous section.

3.3.3 Inter-Tracker Correlation

The second stage of the fusion approach computes a tracker's confidence by assessing its level of congruence with the remaining trackers in the ensemble for a single frame. The individual trackers' credibilities are estimated through an iterative pair-wise correlation procedure among trackers' estimates, as presented in Equation 3-4:

$$C^s_{i,n} = \eta_i C_{i,n} + \frac{1 - \eta_i}{I - 1} \sum_{j=1}^{j=I} \Phi(R_{j,n}, R_{i,n}) C^{s-1}_{i,n} \quad , \forall\, i \neq j \tag{3-4}$$

where, $C^s_{i,n}$ represents the credibility coefficient for the *i-th* tracker after the s iteration, $\eta_i \in [0, 1]$ is a weighting coefficient that controls the importance of the initial credibility $C_{i,n}$, I denotes the total number of trackers, $R_{j,n}$ and $R_{i,n}$ are the tracking estimates at the *n-th* frame for the *i-th* and *j-th* trackers, respectively, and $\Phi(\cdot)$ is a relation coefficient between the *i-th* and *j-th* trackers that measures the spatial congruence, which may be computed using either the $F(\cdot)$ or $r(\cdot)$ similarity metric, as described earlier in this section.

Notice that after convergence the credibility coefficients $C^s_{i,n}$ becomes the final credibility coefficients $C^f_{i,n}$, which will be also used for the *Intra-Tracker Correlation* at the $n + 1$-*th* frame.

3.3.4 Estimates Combination

The last stage in the fusion approach computes the fusion estimate R_{fusion} through a weighted sum of the trackers' estimates R_i, formally:

$$R_{fusion} = \sum_{i=1}^{I} \pi_i R_i \tag{3-5}$$

where the weighting coefficient π_i for the *i-th* tracker is based on the final credibilities coefficients as follows:

$$\pi_i = \frac{C_{i,n}^f}{\sum_{j=1}^{I} C_{j,n}^f} \tag{3-6}$$

3.4 Inspection Module

Trackers are usually prone to drift away from the target face, especially when the target face experiences abrupt changes in appearance. Face detectors, on the other hand, determine accurately the position and dimensions of the bounding box around a face

Thus, the inspection module occasionally exploits the outcome of an offline face detector to provide a potentially more accurate estimate of the state of the target face.

Although offline face detectors have proven to be accurate at locating faces in images, they are inappropriate for the task of tracking due to some operational characteristics: they take considerably more time to process an image compared to face tracking algorithms, they might deliver multiple detection outcomes, and they lack associative capabilities to find the face being tracked among the detection outcomes.

Under the aforementioned circumstances, the Inspection module periodically provides reliable information about the state of the target face. In this manner, it is possible to use the inspection estimate to improve the final tracking estimate, as well as for updating the trackers and preventing possible tracking failures.

The Inspection module checks if the detected face corresponds to the target face. This is done by measuring the dissimilarity between the target face and each of the detector's outcomes. If the lowest computed dissimilarity is below a given threshold the detected face is considered the right one. Otherwise, the Inspector sends no inspection estimate to the Integration module.

In this way, the Inspection module provides an alternative state estimate of the target face that tends to represent a more accurate result than the fusion estimate. These two estimates will be analyzed by the Integration module to compute an improved tracking estimate, which will also be used to update the tracking trajectory or to re-initialize the trackers. Therefore, the final tracking estimate can help to prevent the drifting problem in the long-term tracking, to recover from a tracking loss or to recapture the facial track after a short disappearance of the target face.

3.5 Integration Module

This module receives as input the fusion and the inspection estimates to compute the final tracking estimate, which also corresponds to the framework's output. Moreover, the module activates the update or re-initialization of each tracker in the ensemble by forwarding the final tracking estimate back to the trackers in the Tracking module.

Although it is possible to use different schemes to combine the two inputs, including the usage of a weighting parameter, in this work we assume that the inspection estimate tends to be more reliable than the fusion estimate. Thus, every time the Inspection module provides a valid outcome, the Integration module selects the inspection estimate over its fusion counterpart to produce the final tracking estimate. Otherwise, it holds on to the fusion estimate.

Finally, the estimation cycle finishes when the Tracking module receives the final tracking estimate from the Integration module and applies the proper correction and updates or re-initializes each tracker. It is important to notice that this feedback mechanism (from the Integration module to the Tracking module) is crucial for the framework's execution. The feedback enables a tracker's update or re-initialization to improve the individual trackers' performance and, consequently, the overall tracking accuracy. Either choice is subjected to an evaluation over accumulated records of dissimilarity measures that involve the trackers' estimates and the integration estimate in some consecutive frames (see Section 3.2 for more information).

4
Experimental Design and Results

This chapter presents an experimental analysis of the proposed Collaborative Face Tracking Framework for long-term face tracking.

The first section presents the facial datasets used for the experiments, which comprehend single-face and multiple-face scenarios. Next, we provide information about the performance metric used to evaluate the framework's performance. Then, we detail the prototype implementation of the proposed framework. Finally, we present the adopted protocols and analyze the results of each set of experiments performed in this work.

4.1
Datasets

Throughout the development of this research, we have collected seventeen facial video sequences from several public datasets, comprising a total of forty-one subjects. Furthermore, we have annotated a set of facial references for thirty-nine of them to evaluate the performance of our framework for long-term face tracking.

Given our interest in evaluating the framework's performance in adverse conditions, we prioritized the collection of video sequences containing different conditions, including multiple-faces. Furthermore, we manually annotated the facial references when they were missing or were considered incorrect in such way that most of the visible facial area is enclosed by a bounding box. Thus, we tried to maintain these annotations uniform throughout the datasets by keeping the bounding box centered with the target face: the superior and inferior edges correspond to the middle forehead and the chin, respectively; whereas the left and right edges correspond to the left and right cheekbones, respectively.

The chosen video sequences offer challenging situations designed for face tracking experiments with multiple variations in single-face and multiple-face scenarios, as described next.

– The **TB-Face** dataset is a collection of eleven facial video sequences for single-face tracking. The dataset is a selected subset of the public

available TB-100[1] visual object benchmark dataset proposed by Wu et al. (2013) for the object tracking in general.

The subjects' faces in the TB-Face dataset are subject to occlusions, background clutters, lighting variations, motion blurriness, deformations, in-plane and out-of-plane rotations, and low resolutions. Moreover, the video sequences have between 134 and 892 frames, and the resolution vary from (240x720) to (200x480) pixels.

The TB-100 dataset already includes reference annotations for the target objects in the video sequences. However, the bounding box positions and extents were not considered consistent throughout the dataset, as illustrated by the red bounding box in Figure 4.1. In order to reduce the impact of such variations upon our results, we generated new references for all face images in the TB-Face dataset as depicted in white bounding boxes in Figure 4.1.

Figure 4.1: Differences in the references' annotations from the TB-100 Visual Object Tracking Benchmark Dataset, in red, and the new annotations used for TB-Face Dataset, in white. Each row corresponds to frame samples from different video sequences, from top to bottom: FaceOcc1, FleetFace, and Jumping.

[1]Available in: <`http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html`>. Last accessed: November 18, 2018.

– The **ChokePOINT**[2] dataset proposed by Wong et al. (2011), consists of fifty-four video sequences designed primarily for people identification in real-world scenarios. The video sequences were recorded using a three camera array installed above two indoor gates to record people as they walk through them.

The dataset has twenty-five different subjects in gate one and twenty-nine in gate two. In forty-eight sequences, the subjects walk one after another and in the remaining six the subjects simulate crowded scenarios, which are suitable for face tracking in multiple-face scenarios .

Moreover, the subjects' faces in the dataset experience variations in illumination, deformations, in-plane and out-of-plane rotations, motion blurriness, severe occlusions and resemblance among faces. Furthermore, the video sequences comprise between 757 and 5750 frames, recorded at a frame rate of 30fps and an image resolution of 800x600 pixels.

Since we already have another dataset for single-face scenarios, we are primarily focused on analyzing the framework's performance in multiple-face scenarios. In addition, because the reference annotations are not provided with the dataset, we have selected only a single video sequence (P2E_S5_C2) from the six crowd-like video sequences available. This video sequence is composed of twenty-three subjects and we have manually annotated the positions and extents for each appearing face along the sequence, which are treated individually, as illustrated in Figure 4.2.

– The **LITIV**[3] dataset, proposed by Bouachir and Bilodeau (2015), comprises a set of four facial video sequences recorded in an indoor environment and emulating both single-face (two video sequences) and multiple-face (two video sequences) scenarios for face tracking.

The subjects' faces in the video sequences undergo severe occlusions, background clutter, resemblance among faces, illumination variations, deformations, in-plane, and out-of-plane rotations. The video frames were recorded at 15 fps with 320x240 pixels of resolution. Additionally, each sequence has between 229 to 608 frames.

In addition to the video sequences, the dataset offers reference (position and extent) annotations of a particular face among those present in the

[2]Available in: <`http://arma.sourceforge.net/chokepoint/`>. Last accessed: November 18, 2018.

[3]Available in: <`http://www.polymtl.ca/litiv/en/codes-and-datasets`>. Last accessed: November 18, 2018.

Figure 4.2: Reference annotations for the crowded scenario in the P2E_S5_C2 video sequence from the ChokePOINT Dataset. Each row corresponds to frame samples with annotations of different subjects. From top to bottom rows reference annotations for subjects four, nine and twenty-two, respectively.

video sequences, as depicted in Figure 4.3. Thus, in this dataset, we have only considered this subject as a target.

Figure 4.3: Reference annotations from the LITIV Dataset. First and second row images correspond to sample frames of the target subject in the jp1 and jp2 video sequences, respectively.

– The Multiple Faces[4] dataset built by Maggio et al. (2007), hereafter called **MOTINAS**, is composed of three video sequences with multiple-faces.

The subjects in the video sequences appear and disappear at will from the camera's field of view, and repeatedly occlude each other. Furthermore, the faces in the video sequences present in-plane and out-of-plane rotations, deformations, and movement speed variations. The video sequences have between 488 and 1277 frames, recorded at 25 fps, with an image resolution of 640 x 480 pixels.

Reference annotations are not available with this dataset. So, we have manually annotated the references (position and extent) for all the three faces in the first sequence and for only one face in the two remaining video sequences. Figure 4.4 presents some frame samples with their corresponding annotations.

Figure 4.4: Reference annotations from the MOTINAS Dataset. Each row corresponds to frame samples with annotations for subject one in a different video sequence. From top to bottom: MultiFaceFast, MultiFaceFrontal, and MultiFaceTurning video sequences.

[4]Available in: <`http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`>. Last accessed: November 10, 2018

Table 4.1 summarizes the video sequences selected for the experiments. It offers relevant information such as the datasets each video sequence belongs to, the number of subjects present in the video sequence (NS) and how many of them have reference annotations (NSA), the number of frames (NF), and, finally, the frames resolutions (FR).

Although we have collected seventeen facial video sequences, we have facial references for forty one subjects throughout the video sequences, as it can be seen by summing the NSA column of Table 4.1. It is worth to mention that we are accounting every appearance in a different video as a new subject, since it does not matter which subject it is. In this manner, we could validate the framework's performance over a set of forty one cases for each experiment proposed in this work.

Table 4.1: Video sequences per dataset summary.

DATASET	SEQUENCES	NS	NSA	NF	FR
TB-FACE	BlurFace	1	1	493	640x480
	Boy	1	1	602	640x480
	David	1	1	470	320x240
	David2	1	1	537	320x240
	FaceOcc1	1	1	892	352x288
	FaceOcc2	1	1	812	320x240
	FleetFace	1	1	707	720x480
	Freeman1	1	1	326	360x240
	Jumping	1	1	313	352x288
	Man	1	1	134	241x193
	Trellis	1	1	569	320x240
ChokePOINT	P2E_S5_C2	23	23	808	800x600
LITIV	jp1	4	1	608	320x240
	jp2	4	1	229	320x240
MOTINAS	MultiFaceFast	3	3	488	720x576
	MultiFaceFrontal	4	1	1277	720x576
	MultiFaceTurning	4	1	1007	720x576

NS stands for the number of subjects in a video sequence; NSA refers to the number of subjects with reference annotations; NF indicates the number of frames; and FR stands for the frame resolution.

4.2 Evaluation Metrics

In the reported experiments, we have evaluated the framework's performance based on the area under the curve score (AUC) computed from a set of intersection-over-union error scores ($e_{IoU}(\cdot)$). The AUC measures the framework's performance in a global way, producing a single accuracy score per evaluated video sequence. The $e_{IoU}(\cdot)$, on the other hand, quantifies the

framework's performance in a local fashion. It generates an accuracy score per evaluated video frame.

We chose the $e_{IoU}(\cdot)$ over other evaluation metrics, because it is sensitive to the difference in sizes between a pair of bounding boxes, relating both spatial congruence (position) and spatial extent (size) into a single measure. Essentially, the $e_{IoU}(\cdot)$ measures the dissimilarity between a tracking estimate, R_n, and its corresponding facial reference, R_n^{ref}, at a given frame n. Formally:

$$e_{IoU}(R_n^{ref}, R_n) = 1 - \frac{Area(R_n^{ref} \bigcap R_n)}{Area(R_n^{ref} \bigcup R_n)} \tag{4-1}$$

where, R_n and R_n^{ref} are defined by separate bounding boxes, $Area(\cdot)$ is the area operator, $\bigcap$ and $\bigcup$ represent the intersection and union operators between the pair of bounding boxes, respectively. Notice that $e_{IoU}(\cdot)$ takes values in the range between 0 and 1.

The $e_{IoU}(\cdot)$ is good at quantifying the framework's tracking accuracy given a video frame, however it does not express the framework's behavior throughout the frames in a video sequence. Alternatively, statistics computed from a set of $e_{IoU}(\cdot)$ scores (i.e., mean, median and so on) can be used to depict the framework's performance tendency in a per video sequence evaluation. Nevertheless, these measures do not explicitly tell whether the tracking was successful along the frames of a video sequence or not. A success plot, on the other hand, relates the tracking accuracy with the number of frames in which the target face is considered to be correctly tracked.

Given a video frame, a target face is assumed to be correctly tracked, if the $e_{IoU}(\cdot)$ is below a pre-defined threshold e_0. Thus, the success plot, $p(e_0)$, is built by computing the proportion of frames where the $e_{IoU}(\cdot)$ is below e_0, for different values of e_0, formally:

$$p(e_0) = \frac{|\{e_{IoU} | e_{IoU} < e_0\}|}{NF} \tag{4-2}$$

where, $\{\cdot\}$ is a set of frames within a video sequence satisfying the numerator's condition, $|\cdot|$ represents the cardinality operator of a set and NF stands for the total number of frames in the video sequence.

Thus, the performance metric adopted in this work is computed considering the success plot $p(e_0)$. Specifically, we have used the area under the curve (AUC) defined by the plot of $p(e_0)$ throughout the experiments.

4.3 Framework Prototype

In order to assess the performance of the proposed framework and the ideas presented in this work, we built a prototype according to the guidelines

described in Chapter 3. The prototype was implemented in C++ using some libraries for computer vision and machine learning algorithms such as OpenCV[5] 3.2.0, DLib[6] 19.15, Caffe[7] 2.0 (Jia et al., 2014), as well as Eigen[8] 3.3.4 for linear algebra computations.

The prototype, which is available upon request in the project's[9] website, operates in batch mode and executes the modules within the framework in a sequential manner on an Intel(R) Core(TM) i7-3930K, 3.20GHz CPU machine with 32GB of RAM running Windows 7. A parallel and more efficient version is expected in the near future. Next, we detail the methods that take part in the implemented prototype.

4.3.1 Tracking Module

The ensemble of trackers implemented within the Tracking module comprises five different trackers that operate individually in a sequential manner. We have selected these trackers according to their achieved performances on several tracking evaluations (Smeulders et al., 2014; Wu et al., 2015; Kristan et al., 2016) and to their unique characteristics, which can be exploited by the framework to improve the face tracking performance.

Furthermore, we adjusted the trackers to fit the operating scheme described in Section 3.2: first, the trackers estimate the state of the target face and wait for the Integration module to deliver the final outcome; next, the trackers use the final tracking estimate to correct their tracking states; finally, the trackers update or re-initialize their facial models accordingly to the Integration module command.

Next, we briefly describe the trackers used in our experiments. For more details, please refer to their respective original works.

– *Tracking-Learning-Detection* (**TLD**) is a tracking framework conceived to perform the long-term tracking of arbitrary objects, e.g., faces (Kalal et al., 2012). It combines a tracker with an online detector to acquire and exploit temporal information about the target object, thus, overcoming the possible appearance variations during tracking.

[5]Available in: <`https://opencv.org/`>. Last accessed: November 18, 2018.

[6]Available in: <`http://dlib.net/`>. Last accessed: November 18, 2018.

[7]Available in: <`http://caffe.berkeleyvision.org/`>. Last accessed: November 18, 2018.

[8]Available in: <`http://eigen.tuxfamily.org/index.php?title=Main_Page`>. Last accessed: November 18, 2018.

[9]Available in: <`http://www.lvc.ele.puc-rio.br/projects/FaceTracking/home.html/`>. Last accessed: December 05, 2018.

As the name suggests, TLD decomposes the long-term tracking task into three stages, namely: Tracking, which produces an estimate of the object's state in incoming frames; Learning, that validates the detector's behavior by analyzing the tracking and detection outcomes, moreover, it produces reliable data for detector's training; and Detection, which locates the object in the current frame to correct the tracking trajectory or to re-start the tracker after tracking failure.

– *Kernelized Correlation Filters* (**KCF**), proposed by Henriques et al. (Henriques et al., 2015), is an online generative tracker of arbitrary objects that exploits what the authors call as a Circulant Matrix to extract thousands of training samples in the frequency domain, and uses a linear ridge regression model to capture the appearances of the object.

– *Structured Output Tracking with Kernels* (**STRUCK**), conceived by Hare et al. (Hare et al., 2016), is an online discriminative tracking algorithm that aims to predict the transformation (state) of an object through consecutive frames of a video sequence, while generating training samples. In essence, the STRUCK tracker prioritizes the quality of the training samples over the quantity, which would enable the update of a variant of a Support Vector Machine with reliable data.

– *Locality Sensitive Histograms* (**LSH**), formulated by He et al. (He et al., 2017), is an adaptive tracking algorithm that focuses on handling changes in appearance as a consequence of variations in illumination and occlusions. To this end, LSH uses a floating-point value histogram which accounts for the influence of every pixel in the image over the regions within the target that describe the object's appearance.

– *Fast Compressive Tracker* (**CT**), proposed by Zhang et al. (Zhang et al., 2014), is a discriminative tracking algorithm that uses an adaptive Naive Bayes binary classifier in a low-dimensional subspace to distinguish between the object and the background. CT performs a projection of the high-dimensional feature space on a randomly chosen low-dimensional space which contains sufficient information to reconstruct the original pattern.

Except for the TLD tracker, which we implemented from scratch, we have used publicly available implementations of the KCF[10], STRUCK[11], LSH[12] and

[10]Available in: <`http://www.robots.ox.ac.uk/~joao/circulant/`>. Last accessed: November 25, 2018.

[11]Available in: <`http://www.samhare.net/research/struck`>. Last accessed: November 25, 2018.

[12]Available in: <`http://www.shengfenghe.com/publications.html`>. Last accessed: November 25, 2018.

CT[13] trackers to build the Tracking module.

In the experiments, the trackers were set to operate with their default parameters. Although the Tracking module allows a partial (position) or total (position and extent) correction of the trackers' state estimates, as described in Section 3.2, we commanded the trackers to perform only partial state corrections. We made this decision to establish a uniform operation, since the CT, LSH and STRUCK process a video frame using a single-scale procedure, whereas the KCF and TLD use a multiple-scale scheme.

Regarding the trackers' update and re-initialization procedures, it is possible to define different methods to choose between this two options. In this work, we consider the trackers' recent history to make a decision. So, for each tracker in the ensemble, a dissimilarity score between the tracker estimate and the final tracking estimate is computed according to the Equation 4-1. Next, its respective recent dissimilarity score is stored into its associated *cyclic dissimilarity buffer* in a first in, first out (FIFO) approach. Finally, the average value from its *cyclic dissimilarity buffer* is computed and compared against to a pre-defined re-initialization threshold δ_r to decide whether the tracker should be updated or re-initialized.

By varying the size (s_b) of the *cyclic dissimilarity buffer*, a tracker might be given a chance to correct its behavior (tracker update) before a more drastic resolution (tracker re-initialization) is performed. We have empirically tested different values of s_b, but we only report in this work values of s_b in the set $\{1, 16, 32\}$, since those were considered relevant to the framework performance.

Regarding the re-initialization threshold δ_r, we considered three different scenarios: a conservative, a volatile and a flexible one. In the conservative scenario, a tracker corrects its behavior by updating its facial model in every frame, so δ_r was set to 1.0. Conversely, in the volatile scenario, a tracker gets re-initialized by starting a new facial model in every processed frame. Thus, δ_r was fixed to 0.0. Finally, the flexible scenario offers a tradeoff between a tracker's update and re-initialization. Here, δ_r was set to 0.7 according to empirical experiments.

In case of tracker's update, a tracker proceeds to incorporate information about the target face on the last evaluated frame into its facial model. In case of tracker's re-initialization, a tracker proceeds to erase all the accumulated information about the target face until then. This procedure involves deleting all records in its facial model, clearing its historical behavior in the fusion algorithm (specifically its associated credibility coefficients), and emptying its

[13]Available in: <`http://www4.comp.polyu.edu.hk/~cslzhang/CT/CT.htm`>. Last accessed: November 25, 2018.

cyclic dissimilarity buffer. After this, the tracker starts from scratch with new information about the initial state of the target face, creating a new facial model and re-starting the *cyclic dissimilarity buffer* with information concerning the last evaluated frame. Finally, the Tracking module asks the Fusion module to re-establish the fusion parameters regarding the tracker in the fusion algorithm.

4.3.2 Fusion Module

The Fusion module is built upon the Symbiotic Tracking Ensemble proposed by Gao et al. (2014). The Fusion module merges the five tracking estimates from the Tracking module into a single outcome.

The relation coefficients $\Theta(\cdot)$ and $\Phi(\cdot)$ described in Section 3.3, which measure the temporal and spatial congruence in the intra-tracker and inter-tracker correlation stages, respectively, allow four variants (rr, rF, Fr, and FF) of Gao's approach, regarding the similarity metrics of Equations 3-1 and 3-2. In our experiments, however, we used the FF variant, which has shown the best fusion results according to the authors' report in visual object tracking tasks.

Furthermore, we followed the Gao's recommendations to configure the remaining fusion parameters. So, for all the participant trackers, their associated prior credibilities C_i were set to 1 and their regularization parameters ξ_i and η_i were set to 0.1 and 0.1, respectively. Thus, we consider in our experiments all the trackers equally reliable.

4.3.3 Inspection Module

We built the Inspection module to support the cooperation between an offline face detector and a face validation algorithm according to the guidelines in Section 3.4.

Although the Inspection module accepts any offline face detector to be part of the framework, we chose to work with the *MTCNN Face Detection and Alignment* (**MTCNN**) algorithm, proposed by Zhang et al. (2016), because it is robust to different imaging conditions, occlusions, and large head posture variations. Furthermore, it provides a low rate of false face detections.

The MTCNN conducts face detection and alignment to boost detection performance. The face detector is organized in a three-stage Convolutional Neural Network. The first two stages deliver candidate regions with a high

probability of containing a face. The third stage performs additional face landmark localization to reduce false positives detections.

Several implementations of the MTCNN are available on the internet, including in the authors' homepage. We used a C++ implementation[14] due to the programming language compatibility with the framework.

As for the face validation algorithm, we employed a variant of the ResNet network proposed by He et al. (2016), which is a Deep Convolutional Neural Network architecture designed for image recognition, named **DNN-Face**. The DNN-Face, which is available[15] within DLib's library, was conceived to group facial images that belong to a subject into a hypersphere of a certain radius (set to 0.6 in this case). The DNN-Face's deep architecture considers twenty-nine out of the thirty-four convolutional layers that constitute the ResNet network.

In our work, the DNN-Face extracts representative descriptors from facial images, which are used to compute the dissimilarity scores and find the target face in the detection outcomes, as described in Section 3.4. So, given a reference facial image that corresponds to the target face collected at the beginning of the tracking task, the process of retrieving the target face from the pool of detections goes as follows. First, the Inspection module normalizes the faces by aligning them to a standard pose and resampling them to a fixed image size of 150x150 pixels. Next, it extracts features from all the normalized faces. Finally, it computes the dissimilarity scores by measuring the distances between the reference face image and each of the detections in the feature space.

The valid face, which should correspond to the target face, is the detection with the lowest dissimilarity score, which must be below a predefined threshold. In this work we set this threshold to 0.6 based on empirical experiments.

In this research, we have also evaluated the impact of the Inspection module over the Fusion and Tracking modules by restricting the use of the MTCNN face detector to every k frames. In our experiments k took values between 1 and 64. In this manuscript, we report the results for those values of k that have shown a significant difference in the final tracking performance, which are 1, 16 and 32.

4.3.4

[14]Available in: <`https://github.com/wowo200/MTCNN`>. Last accessed: Novemeber 18, 2018.

[15]Available in: <`https://github.com/davisking/dlib/blob/master/examples/dnn_face_recognition_ex.cpp`>. Last accessed: November 18, 2018.

Integration Module

As already mentioned in Section 3.5, the Integration module may adopt different ways to combine the fusion estimate, provided by the Fusion module, with the inspector estimate, supplied by the Inspection module, to produce the final tracking estimate.

Throughout this research, we have assumed that the Inspection module produces more reliable estimates than the Fusion module. Thus, if the Inspection module is active and provides a valid estimate of the target face state (inspection estimate), then the Integration module sets the the inspection estimate as the final tracking estimate. Otherwise, the Integration module sets the fusion estimate as the final estimate.

Finally, the Integration module forwards the final tracking estimate back to the Tracking module, so the trackers may update or re-initialize their facial models.

4.4 Experimental Protocol

4.4.1 Framework's Configurations

Next, we present the set of framework's configurations tested in our experiments.

- *inspection-only*: in this configuration, there are no trackers involved, but just a detector operating within the Inspection module. So, its outcome, when produced, is supposed to correspond to the target face. In other words, this configuration puts the Inspection module to work on the face tracking task. We simulated this configuration by activating the Inspection and Integration modules in our framework, as depicted in Figure 4.5(a).
- *single tracking*: this is the stand-alone tracker solution. This configuration was simulated by enabling the Tracking, Fusion and Integration modules in our framework. However, we only activate a single tracker within the Tracking module, as shown in Figure 4.5(b), which makes the final tracking estimate to correspond to the tracking estimate.
- *tracking ensemble*: here, the outcomes of a committee of trackers are consolidated into a single consensus outcome. This configuration, which basically corresponds to the Gao's fusion approach, was obtained by enabling the Tracking, Fusion and Integration modules in our framework,

as illustrated in Figure 4.6(a). In this configuration, the Tracking module comprises multiple trackers. Moreover, Integration does not feedback the final estimate.

– *tracking ensemble + feedback*: this setting corresponds to our prior work (Ayma et al., 2017), which aimed at improving the fusion method proposed by Gao et al. through the addition of the feedback process. So, in this configuration, the Integration module disregards the Inspector output and forwards the fusion estimate back to the Tracking module. This setting was obtained by activating the Tracking, Fusion and Integration modules in our framework, as well as the feedback mechanism, as shown in Figure 4.6(b).

– *collaborative single tracking*: in this case, a single tracker benefits from the framework structure, which includes the face detector present in the Inspection module. We simulated this configuration by activating all the modules and the feedback mechanism in our framework, whereby just a single tracker is enabled in the Tracking module, as depicted in Figure 4.7(a). It basically corresponds to the whole framework, but with only a single tracker being used.

– *collaborative tracking*: this setting corresponds to the entire framework operation, as described in the foregoing sections. In this configuration, the Fusion module consolidates the outcomes of multiple tackers into a single consensus outcome, which is replaced by the inspection estimate whenever the Inspection module retrieves the target face from the set of detections produced by the face detector. The Integration module forwards the final tracking estimate to the Tracking module to either updating or re-initializing the trackers. We simulated this configuration by enabling all the components in the framework, as depicted in Figure 4.7(b).

Notice that the trackers are either updated or re-initialized according to the re-initialization threshold δ_r described in Section 3.5. Furthermore, as established in Section 4.3.1, δ_r took the following values 0.0, 0.7 and 1.0, which enabled three variants in both the *collaborative individual tracking* and the *collaborative tracking* configurations.

In the first variant, the volatile one ($\delta_r = 0.0$), the trackers always get re-initialize. Conversely, in the second variant, which corresponds to the conservative one ($\delta_r = 1.0$), the trackers are always forced to update their models. Finally, the last variant, called hereafter flexible ($\delta_r = 0.7$), allows a tradeoff between trackers' updating and re-initialization.

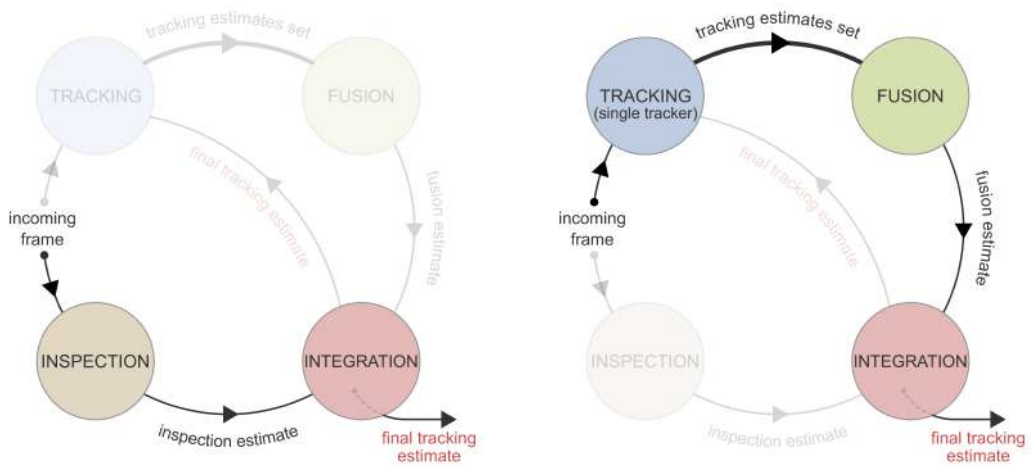


Figure 4.5: Collaborative Face Tracking: *inspection–only* (a) and *single tracking* (b) configurations.

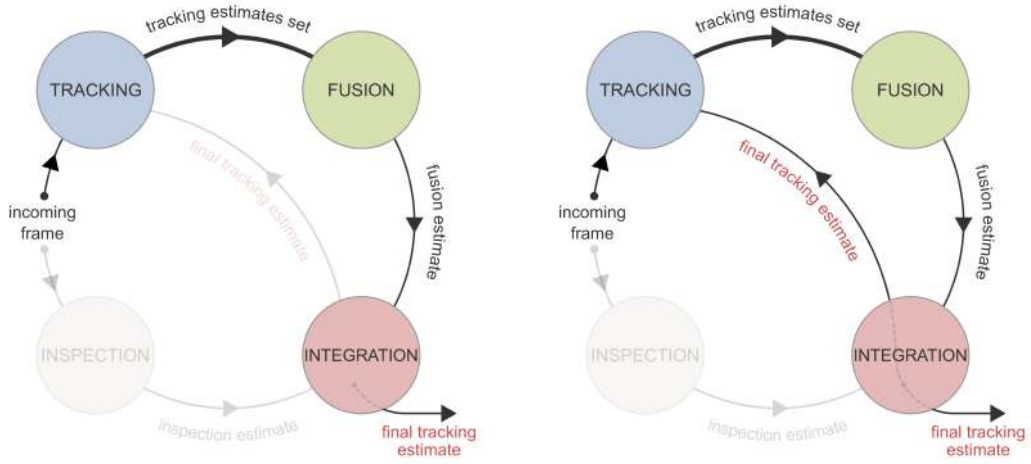


Figure 4.6: Collaborative Face Tracking: *tracking ensemble* (a) and *tracking ensemble + feedback* (b) configurations.

4.4.2 Experiments

Based on the aforementioned framework's configurations, we have designed three main experiments to assess the framework's performance in adverse conditions, especially in single-face and multiple-face scenarios, including occlusions.

– *On the Collaborative Tracking – Inspection Module Validation*: this experiment aims to assess the Inspection module ability to find and validate the target face among the outcomes provided by the face detector.

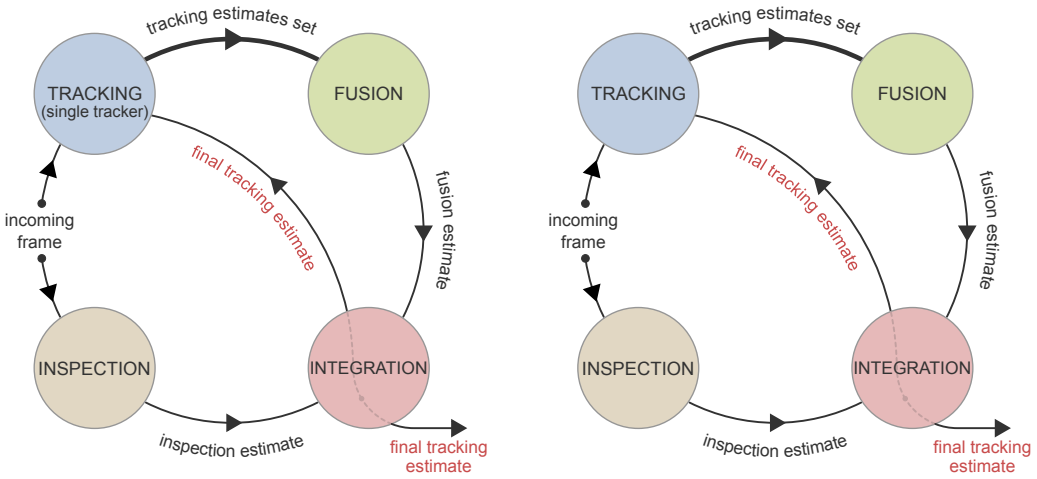


Figure 4.7: Collaborative Face Tracking: *collaborative single tracking* (a) and *collaborative tracking* (b) configurations.

In this experiment, the framework's prototype operates according to the *inspection-only* configuration. Furthermore, the Inspection module should continuously intervene in the framework's prototype execution for its proper evaluation, so we set the frequency of the Inspector operation $k = 1$.

– *On the Collaborative Tracking with a Single Tracker*: this experiment aims to evaluate the impact of incorporating information provided by the face detector about the target's face state into a tracker's facial model.

  Here, the framework's prototype operates according to three configurations: *inspection-only*, *single tracking* and *collaborative single tracking*. The latter was applied with its three variants: *volatile*, *conservative* and *flexible*. Moreover, to assess the influence of the face detector on a tracking algorithm, the frequency of the Inspection module operation was varied from continuous to sporadically, by setting $k = 1$ and $k = 32$, respectively.

– *On the Collaborative Tracking with an Ensemble of Trackers*: this experiment aims to assess the framework performance by complementing the consensus output of the Fusion module with information delivered by the face detector in the Inspection module.

  In this experiment, the framework's prototype operates according to the *single tracking*, *tracking ensemble* and *tracking ensemble + feedback* configurations, as well as the *collaborative tracking* configuration in its *volatile*, *conservative* and *flexible* variants. Furthermore, the Inspection module operates in a continuous, periodic and sporadically ways, i.e., in

every $k = 1$, $k = 16$ and $k = 32$ frames, and the size of the trackers' *cyclic dissimilarity buffers* varies according to $s_b = 1$, $s_b = 16$ and $s_b = 32$ positions-length.

4.5 Results

This section provides an analysis of the framework's performance on the face tracking problem regarding the results obtained at executing the framework's prototype according to the experimental protocol described in the foregoing sections of this Chapter.

In all the experiments, the tracking process started with the first bounding box found by the face detector, which corresponds to the initial state of the desired target face in a given video sequence. Furthermore, this bounding box is used by the Inspection module to collect the reference facial image for the face validation procedure.

We performed the analysis of the results in each experiment from two perspectives which relate to the single-face and multiple-face scenarios. Moreover, in agreement with the reference annotations, we only considered the frames containing a visible target face for evaluating the framework's performance.

4.5.1 On the Collaborative Tracking – Inspection Module Validation

In this experiment, we expressed the Inspection module performance in terms of the correct and incorrect responses rates, which relate to the hit and miss occurrences of the target face. A correct response occurs when the Inspection module provides a state estimate that corresponds with the target face state, i.e., position and extent, in a given video frame. An incorrect response, on the other hand, takes place when the Inspection module delivers a non-related outcome to the target face state, which might occur in two ways: the Inspection module provides a state estimate that does not correspond to the target face, or it does not provide a state estimate at all (non-response).

Tables 4.2 and 4.3 summarize the Inspection module performance for single-face and multiple-face video sequences, respectively. Both tables present the correct response rate (CRR) and the two possible incorrect responses rates that result from the erroneous state estimate (ERR) and the non-response rate (NRR), along with the number of frames where the target face remains visible (VF). Furthermore, we included the non-detection rate of the face detector (NDR), which is a related to NRR: while NRR refers to the Inspection Module,

NDR is related to the face detector. Thus, the NRR comprehends the cases where the face detector does not detect any face (NDR) and the cases where none of the detected faces were considered valid. Moreover, for the sequences with multiple faces in Table 4.3, we concatenated the number of the respective subject next to the sequences' names.

Table 4.2 shows that for the single-face video sequences, the Inspection module often produced a state estimate that matched the target face state, scoring an CRR of 0.82 on average. Additionally, the EER was 0 for all the sequences, showing that the Inspector module does not provide any wrong estimate to the framework in the single-face scenario.

Table 4.2: Inspection module performance's summary for datasets whose video sequences contain a single face.

DATASET	SEQUENCE	VF	CRR	ERR	NRR	NDR
TB-FACE	BlurFace	493	0.99	0.00	0.01	0.00
	Boy	602	0.63	0.00	0.37	0.02
	David	471	0.91	0.00	0.09	0.00
	David2	537	0.81	0.00	0.19	0.18
	FaceOcc1	892	0.82	0.00	0.18	0.14
	FaceOcc2	812	0.56	0.00	0.44	0.43
	FleetFace	707	0.77	0.00	0.23	0.19
	Freeman1	326	0.78	0.00	0.22	0.20
	Jumping	313	0.88	0.00	0.12	0.09
	Man	134	0.99	0.00	0.01	0.00
	Trellis	569	0.88	0.00	0.12	0.07
	Average		0.82	0.00	0.18	0.12

CRR, ERR and NRR stands for the Inspection module correct response rate, erroneous response rate and non-response rate, respectively; NDR represents the non-detection rate of the face detector; and VF refers to the number of video frames where the target face remains visible.

Despite the Inspection module overall good performance, it achieved a CRR of only 0.63 and 0.56 for the Boy and FaceOcc2 video sequences, respectively. The Inspection module performance for the FaceOcc2 video sequence is a result of a non-response by the face detector (NDR of 0.43), i.e., the face detector did not find any faces in almost half of the frames due to the severe occlusions and large head posture variations experienced by the target face. Figure 4.8 shows some video frames in which the face detector failed at locating faces, producing an Inspection module failure. On the other hand, the low performance in the Boy video sequence is mostly related to the face validation procedure in the Inspection module, which was not able to validate the target face in most of the video frames (NRR of 0.37) despite the face detector good performance (NDR of 0.02). Figure 4.9 presents some video frames examples in which the face validation algorithm failed at validating

the target face from the set of detection outcomes illustrated by the bounding boxes in yellow.

Figure 4.8: Video frames examples of the FaceOcc2 sequence in which the face detector did not provide any outcome, producing a non-response by the Inspection module.

Figure 4.9: Video frames examples in which the face validation algorithm in the Inspection module failed at validating the target face in the Boy video sequence. The bounding boxes in yellow depict the face detection outcomes.

Table 4.3 shows that the Inspection module was capable of accurately deliver an inspection estimate that corresponded to the target face in the

MOTINAS and LITIV datasets achieving a CRR near to 1.0. It is also worth to note that the Inspection module did not deliver any wrong estimate for these datasets, since the ERR was equal to 0.

Table 4.3: Inspection module performance's summary for datasets whose video sequences contain multiple faces.

DATASET	SEQUENCE	VF	CRR	ERR	NRR	NDR
MOTINAS	MultiFaceFast_1	356	0.99	0.00	0.01	0.00
	MultiFaceFast_2	394	0.96	0.00	0.04	0.00
	MultiFaceFast_3	444	0.99	0.00	0.01	0.00
	MultiFaceFrontal_1	958	0.98	0.00	0.02	0.00
	MultiFaceTurning_1	778	0.91	0.00	0.09	0.00
	Average		0.97	0.00	0.03	0.00
LITIV	jp1_1	551	0.88	0.00	0.12	0.00
	jp2_1	216	0.99	0.00	0.01	0.00
	Average		0.93	0.00	0.07	0.00
ChokePOINT	P2E_S5_C2_1	128	0.90	0.05	0.05	0.00
	P2E_S5_C2_2	103	0.71	0.20	0.09	0.00
	P2E_S5_C2_3	66	0.92	0.00	0.08	0.00
	P2E_S5_C2_4	135	0.56	0.18	0.27	0.00
	P2E_S5_C2_5	51	0.53	0.04	0.43	0.00
	P2E_S5_C2_6	90	0.98	0.02	0.00	0.00
	P2E_S5_C2_7	146	0.55	0.26	0.19	0.00
	P2E_S5_C2_8	130	0.15	0.50	0.35	0.00
	P2E_S5_C2_9	124	0.55	0.28	0.17	0.00
	P2E_S5_C2_10	117	0.41	0.20	0.39	0.00
	P2E_S5_C2_11	135	0.76	0.04	0.20	0.00
	P2E_S5_C2_12	169	0.37	0.43	0.20	0.00
	P2E_S5_C2_13	134	0.78	0.13	0.09	0.00
	P2E_S5_C2_14	190	0.42	0.42	0.17	0.00
	P2E_S5_C2_15	115	0.65	0.11	0.23	0.00
	P2E_S5_C2_16	161	0.67	0.27	0.06	0.00
	P2E_S5_C2_17	103	0.95	0.03	0.02	0.00
	P2E_S5_C2_18	120	0.45	0.15	0.40	0.00
	P2E_S5_C2_19	148	0.17	0.38	0.45	0.00
	P2E_S5_C2_20	204	0.97	0.00	0.03	0.00
	P2E_S5_C2_21	142	0.94	0.02	0.04	0.00
	P2E_S5_C2_22	183	0.70	0.03	0.27	0.00
	P2E_S5_C2_23	143	0.63	0.00	0.37	0.00
	Average		0.64	0.16	0.20	0.00

CRR, ERR and NRR stands for the Inspection module correct response rate, erroneous response rate and non-response rate, respectively; NDR represents the non-detection rate of the face detector; and VF refers to the number of video frames where the target face remains visible.

Considering the ChokePOINT dataset, the Inspection module performed significantly worse achieving an average CRR of 0.64. The worse cases were for subjects 8 (CRR of 0.15) and 19 (CRR of 0.17). In average, the estimates

were incorrectly delivered for 16% of the frames (ERR of 0.16) while in 20% of the frames, all the detections were considered invalid (NRR = 0.20 and NDR = 0.00). We believe that this poor performance was related to the reference facial images collected at the beginning of the tracking process and used for the face validation procedure of the detection outcomes. In the MOTINAS and LITIV datasets, the subjects presented facial images in an approximately frontal pose with good resolution at the beginning of the tracking process, as illustrated in Figure 4.10(a) and Figure 4.10(b), respectively. The subjects from the ChokePOINT dataset, on the other hand, presented facial images with poorer resolution. Furthermore, just a few of them had facial images in a nearly frontal posture at the beginning of the tracking, as it can be observed in Figure 4.11. Since the Figure presents the subjects sorted descendingly according to the CRR scores in Table 4.3, it is possible to note some correspondence trend between higher CRR scores and near-frontal face positions.

(a) (b)

Figure 4.10: Reference templates per video sequences, containing good resolution facial images in nearly frontal postures of subjects from the MOTINAS dataset (a) and the LITIV dataset (b).

Figure 4.11: Reference images, containing facial images of the subjects from the ChokePOINT dataset with poorly resolutions. From the top-left to the bottom-right, the subjects (numbers in white) are sorted descendingly according to the CRR score in Table 4.3 which tends to correspond to the changing from near frontal postures.

In this sense, intrigued by the poor performance shown by the Inspection module over the ChokePOINT dataset, we conducted a test to check if our

assumption was correct. Thus, we investigated the impact of using only near frontal faces as the reference facial images for the face validation procedure. To this end, we executed another experiment for the subjects with associated CRR scores below to 0.6, specifically subjects 4, 5, 7, 8, 9, 10, 12, 14, 18 and 19. In this new experiment, the tracking process was only started when those subjects presented facial images in a near frontal posture. Visual examples are shown in Figure 4.12, where it is possible to compare the new and the original reference facial images for the validation procedure. It is important to mention that this selection was made through visual analysis and led to a reduction in the number of available frames of the video sequences corresponding to those subjects.

(a)

(b)

Figure 4.12: Reference facial images of the subjects (numbers in white) from the ChokePOINT dataset: (a) shows profile facial images; (b) presents facial images in a near frontal posture.

Similarly to Table 4.3, Table 4.4 summarizes the performance of the Inspection module for the ChokePOINT dataset concerning the selected subjects in the new experiment, called FRONTAL REFERENCE. By examining Table 4.4, it is possible to notice a significant improvement of 0.34 in CRR average score at validating the target faces. A rigorous evaluation of the table regarding subjects 8 and 19, which were the worse cases, depicts a gain in performance of 0.43 and 0.75 over the original 0.15 and 0.17, respectively. Furthermore, subject 5 presented a perfect CRR score, with a gain of 0.47 over the original 0.53 score, but for almost half of the evaluated video frames. Moreover, the ERR went from 0.28 to 0.09, reducing by one-third the amount of incorrect estimates.

Table 4.4: Inspection module performance's summary for the ChokePOINT dataset. The table presents the performance results related to the subjects whose facial images in the reference templates (ORIGINAL) were replaced with facial images having a nearly frontal posture (FRONTAL REFERENCE).

SEQUENCE	ORIGINAL				FRONTAL REFERENCE			
	VF	CRR	ERR	NRR	VF	CRR	ERR	NRR
P2E_S5_C2_4	135	0.56	0.18	0.27	105	0.80	0.18	0.02
P2E_S5_C2_5	51	0.53	0.04	0.43	23	1.00	0.00	0.00
P2E_S5_C2_7	146	0.55	0.26	0.19	137	0.76	0.08	0.16
P2E_S5_C2_8	130	0.15	0.50	0.35	115	0.58	0.20	0.22
P2E_S5_C2_9	124	0.55	0.28	0.17	101	0.63	0.16	0.21
P2E_S5_C2_10	117	0.41	0.20	0.39	68	0.79	0.00	0.21
P2E_S5_C2_12	169	0.37	0.43	0.20	137	0.60	0.26	0.15
P2E_S5_C2_14	190	0.42	0.42	0.17	155	0.66	0.05	0.30
P2E_S5_C2_18	120	0.45	0.15	0.40	96	0.85	0.00	0.15
P2E_S5_C2_19	148	0.17	0.38	0.45	125	0.92	0.02	0.06
Average		0.42	0.28	0.30		0.76	0.09	0.15

CRR, ERR and NRR stands for the Inspection module correct response rate, erroneous response rate and non-response rate, respectively; NDR represents the non-detection rate of the face detector; and VF refers to the number of video frames where the target face remains visible.

These results show that the face detector performs remarkably well at locating faces; however, it fails when the faces undergo severe occlusions and/or large head posture variations. The results tend to indicate that the Inspection module performs considerably well at finding and validating the target face from the set of detection outcomes delivered by the face detector. However, the target face validation procedure strongly depends on the reference facial image used for measuring the similarity with the detection outcomes. The results also suggest that the resolution of the facial images might drastically affect the performance of the target face validation procedure, hence, the Inspection Module performance as well. Furthermore, the results evidence a strong dependence on the facial images postures: the more frontal is the posture in the reference facial image, the higher the chances that the Inspection module finds the target face.

4.5.2 On the Collaborative Tracking with a Single Tracker

As described in Section 4.4.1, this experiment aimed to evaluate a tracker's behavior at considering information from the face detector during the tracking process. Here, we compared the average $AUC(e_{IoU})$ scores of the *inspection-only* and *single tracking* configurations, as well as the average $AUC(e_{IoU})$ scores of the three variants of the cooperation between a tracker

and the face detector, which are the *volatile*, *conservative* and *flexible* variants of the *collaborative single tracking*.

Figure 4.13 exhibits the framework's performance expressed in terms of the average $AUC(e_{IoU})$ scores for the single-face video sequences from the TB-Face dataset. In Figure 4.13(a), the *inspection-only* and *collaborative single tracking* configurations results correspond to the framework's execution with a continuous Inspection module participation ($k = 1$) and a *cyclic dissimilarity buffer* of size $s_b = 1$. The figure shows that each tracker in the *single tracking* configuration presents a different performance. Except for the KCF, the remaining trackers obtain inferior scores than the *inspection-only* configuration. Moreover, all the three variants of the *collaborative single tracking* configuration performed similarly and outperformed both the *inspection-only* and *tracking individual* configurations.

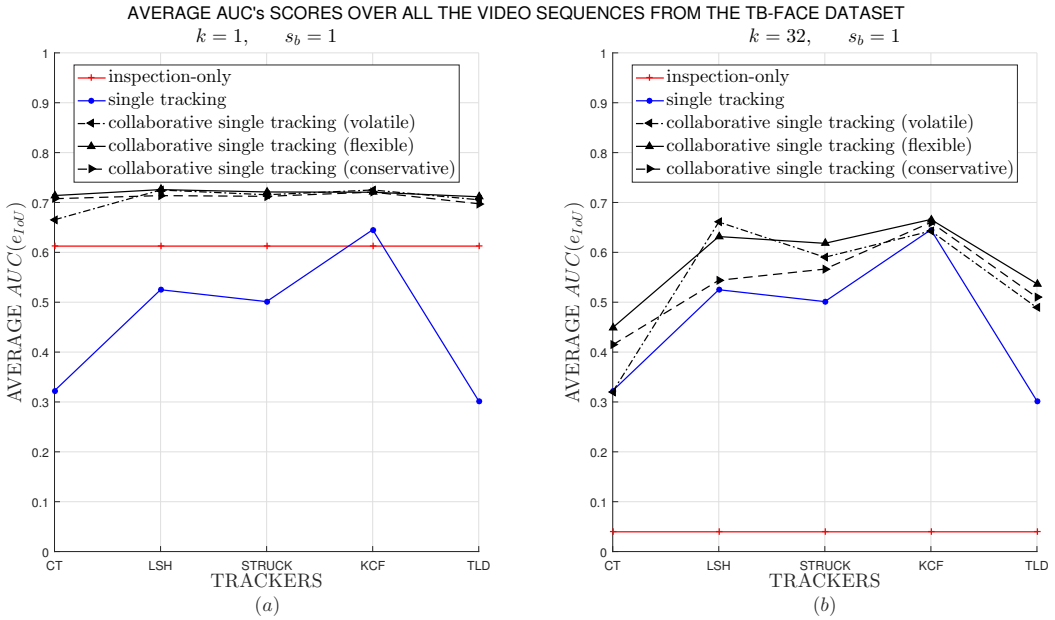


Figure 4.13: Framework's performance on single-face video sequences from the TB-Face dataset. The graphics present the average $AUC(e_{IoU})$ scores obtained by combining the face detector with each tracker for an intervention of the Inspection module at every $k = 1$ (a) and $k = 32$ (b) frames.

On the other hand, the *inspection-only* and the *collaborative single tracking* configurations results, shown in Figure 4.13(b), are consequence of Inspection module participation at only every $k = 32$ frames. This makes the *inspection-only* configuration to present poor results, since in most of the frames it does not deliver any outcome. The three variants of the *collaborative single tracking* also present a drop in performance when compared to the $k = 1$ experiment (Figure 4.13(a)), however it still perform better than the trackers in the *tracking individual* configuration, showing that even with a reduced

participation it can improve the tracking performance. Another observation is the variation of the *collaborative individual tracking* variants which has a similar trend as the single trackers, as expected given the intermittent participation of the Inspection module. Furthermore, the flexible variant outperforms the other two variants.

Figure 4.14 shows the framework's performance based on the average $AUC(e_{IoU})$ scores for video sequences containing multiple faces from the MOTINAS dataset. Figure 4.14(a), exhibits the results of the *inspection-only* and *collaborative single tracking* configurations, obtained by executing the Inspection module at every frame. The figure shows that three variants of the *collaborative single tracking* configuration performed similarly, and considerably outperformed the *single tracking* configuration. Moreover, the *inspection-only* configuration behaved slightly inferior to the three *collaborative single tracking* configuration variants.

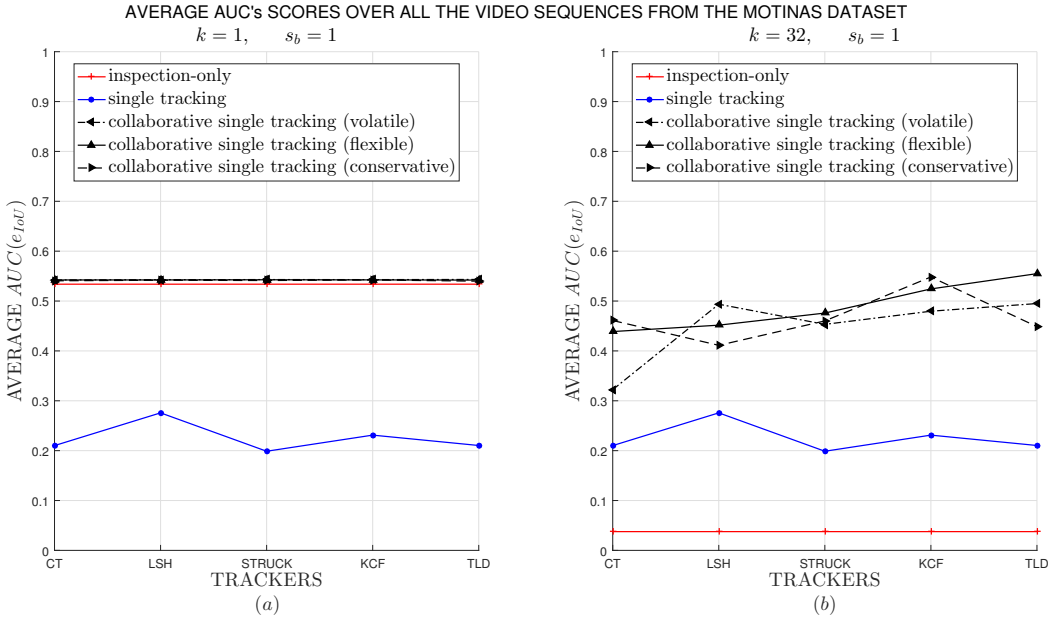


Figure 4.14: Framework's performance on multiple-face video sequences from the MOTINAS dataset. The graphics present the average $AUC(e_{IoU})$ scores obtained by combining the face detector with each tracker for an intervention of the Inspection module at every $k = 1$ (a) and $k = 32$ (b) frames.

By limiting the Inspection module operation frequency to every $k = 32$ frames, the *collaborative single tracking* configuration presented a drop in performance, as illustrated in Figure 4.14(b). Nevertheless, the three variants of the *collaborative single tracking* configuration were superior to the *single tracking* and to the *inspection-only* configurations. The latter presented a very low performance, because in most of the frames the Inspection module did not deliver any estimate due to its low operation frequency. We can also note that

the flexible variant presented a slightly better performance, in average, than the other two variants,

Figures 4.13(a) and 4.14(a) show that the *collaborative single tracking* configuration performed better than the *inspection-only* configuration in the TB-Face and MOTINAS datasets, respectively. Indeed, the gain in performance for the TB-Face dataset that relates both configurations is significantly greater than the gain in performance for the MOTINAS dataset. This difference in the performances' gains is a consequence of the non-response by the Inspection module, which according to Tables 4.2 and 4.3 are of 18% and 3% of the times for the TB-Face and MOTINAS datasets, respectively. This means that the trackers benefit from the face detector in almost all the processed video frames in the MOTINAS dataset.

These results indicate that the framework and the individual tracking algorithms benefit from the information provided by the face detector, which comes from the Inspection module. In fact, through the analysis of the low frequency Inspection module operation ($k = 32$), it is possible to note that the feedback present in the *collaborative single tracking* configuration is working and it enhances the trackers' performances, since their average $AUC(e_{IoU})$ scores are considerable superior than the other configurations. Moreover, the more the Inspection module participates in the framework's execution, the better and more consistent is the overall tracking performance among the individual trackers.

Is worth mentioning that the Inspection module supports the Tracking module by providing information about the target face state coming from the face detector; this information is used by the Tracking module to improve the trackers.

4.5.3 On the Collaborative Tracking with an Ensemble of Trackers

This experiment aimed to assess the framework's performance with all of its features enabled, as described in Section 4.4.1. Here, we compared the average $AUC(e_{IoU})$ scores of the volatile, conservative and flexible variants of the *collaborative tracking* configuration, which correspond to the framework's complete solution, against the *tracking ensemble*, *tracking ensemble + feedback* and the *single tracking* configurations.

Figure 4.15 presents the average $AUC(e_{IoU})$ scores for the video sequences containing single-faces from the TB-Face dataset, while Figures 4.16, 4.17 and 4.18 present the results on the video sequences containing multiple-faces, considering LITIV, MOTINAS, and ChokePOINT datasets, respectively.

Although we preliminary conducted several experiments varying the tracker's *cyclic dissimilarity buffer* size (s_b), in accordance to Section 4.4.1, on the following analysis we present only the results that correspond to $s_b = 1$ and $s_b = 32$, as there is no significative difference in using intermediary s_b values.

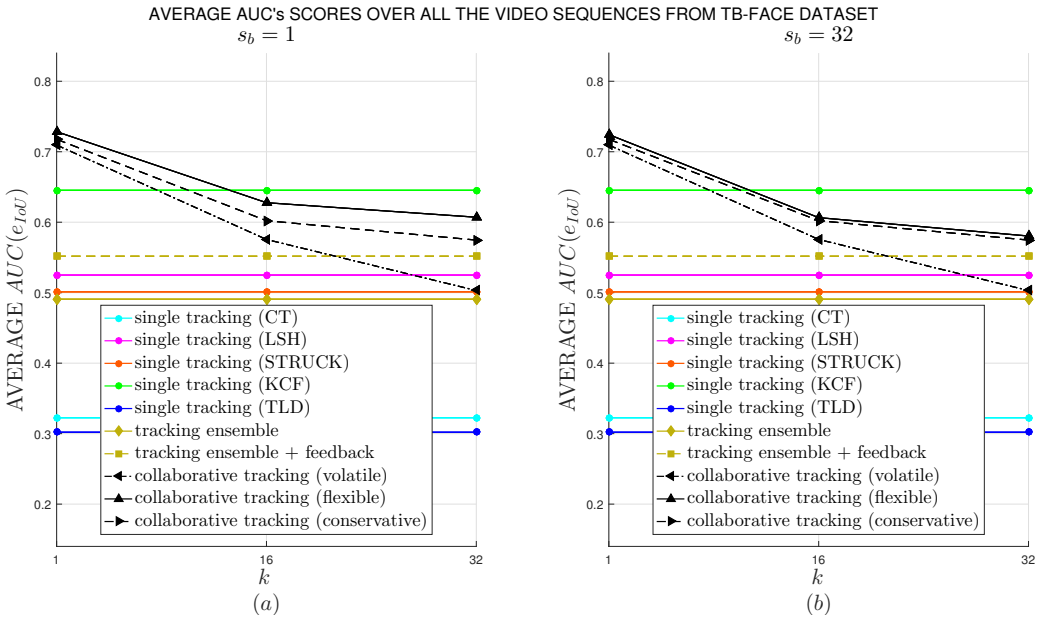


Figure 4.15: Framework's performance on single-face video sequences from the TB-Face Dataset. The graphics present the $AUC(e_{IoU})$ results obtained by combining trackers and complementing their fusion with a face detector; all of them for an intervention of the Inspector module at every $k = 1$, $k = 16$ and $k = 32$ frames, and a re-initialization buffer of size $s_b = 1$ (a) and $s_b = 32$ (b).

In all cases, the *tracking ensemble + feedback* configuration achieved better performance than its *tracking ensemble* counterpart, showing the importance of the feedback mechanism in the framework. Nevertheless, the *tracking ensemble + feedback* configuration was worse than the best performing individual tracker, except for the MOTINAS dataset. Although these results suggest that resubmitting the fusion estimate back to the trackers might be beneficial for the face tracking in general, the bare trackers' updating with information delivered by the tracking fusion algorithm might not be enough to build robust facial models within the trackers: some of the trackers might still perform poorly, negatively influencing the fusion process.

The results also confirm that the inclusion of the face detector brought substantial performance gains. We observe that the *collaborative tracking* configuration performed better, if not similar, to their *tracking ensemble + feedback* counterpart. However, the performance of its three variants tends to decrease as the participation of the Inspection module gets restricted, by setting $k = 16$ and $k = 32$. Interestingly, in most of the datasets, the *volatile* variant showed to be inferior to the *conservative* variant, whereas the *flexible*

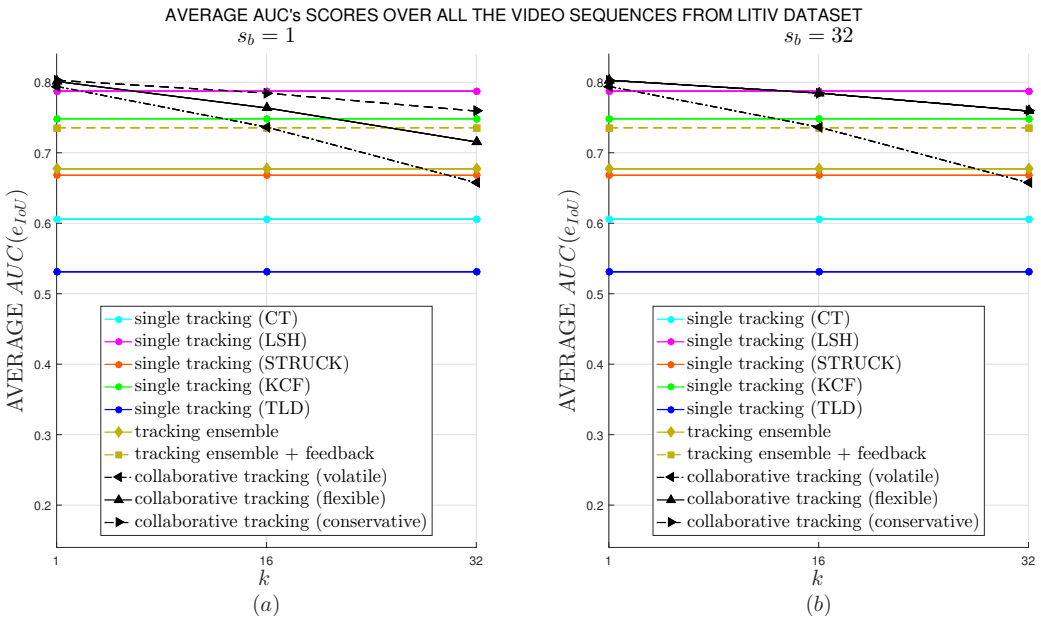


Figure 4.16: Framework's performance on multiple-face video sequences from the LITIV Dataset. The graphics present the $AUC(e_{IoU})$ results obtained by combining trackers and complementing their fusion with a face detector; all of them for an intervention of the Inspector module at every $k = 1$, $k = 16$ and $k = 32$ frames, and a re-initialization buffer of size $s_b = 1$ (a) and $s_b = 32$ (b).

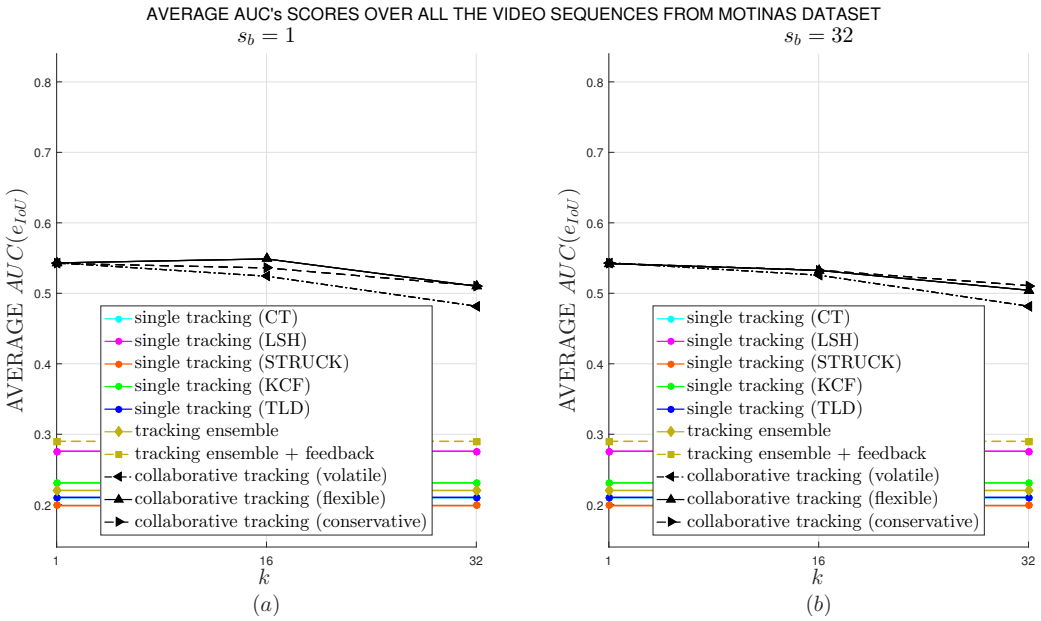


Figure 4.17: Framework's performance on multiple-face video sequences from the MOTINAS Dataset. The graphics present the $AUC(e_{IoU})$ results obtained by combining trackers and complementing their fusion with a face detector; all of them for an intervention of the Inspector module at every $k = 1$, $k = 16$ and $k = 32$ frames, and a re-initialization buffer of size $s_b = 1$ (a) and $s_b = 32$ (b).

variant was the best among them. The difference in performance among these variants might be related to the trackers' re-initialization procedure, regarding

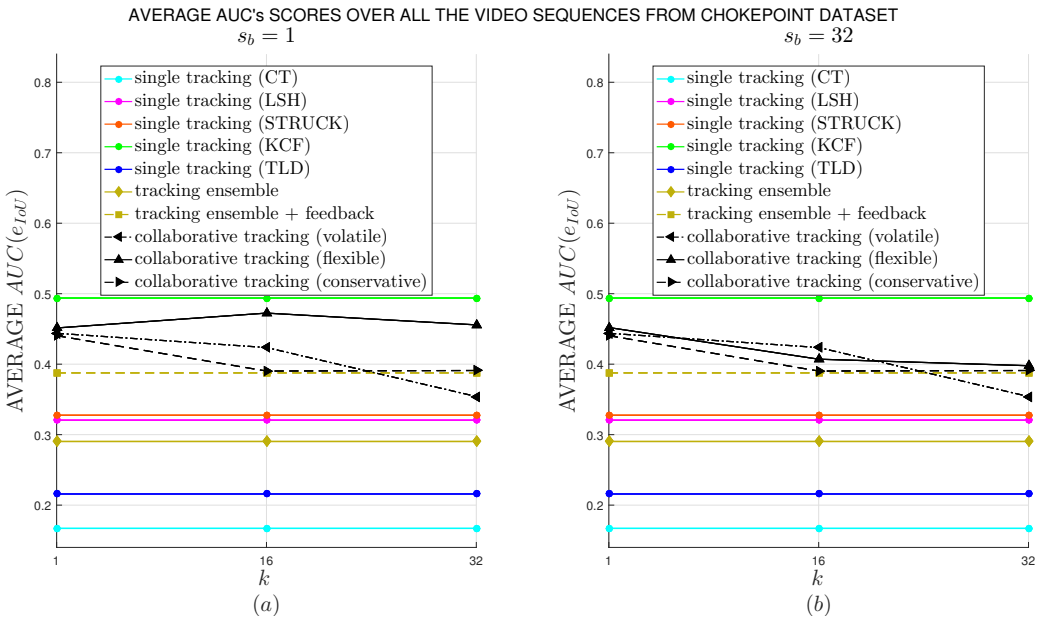


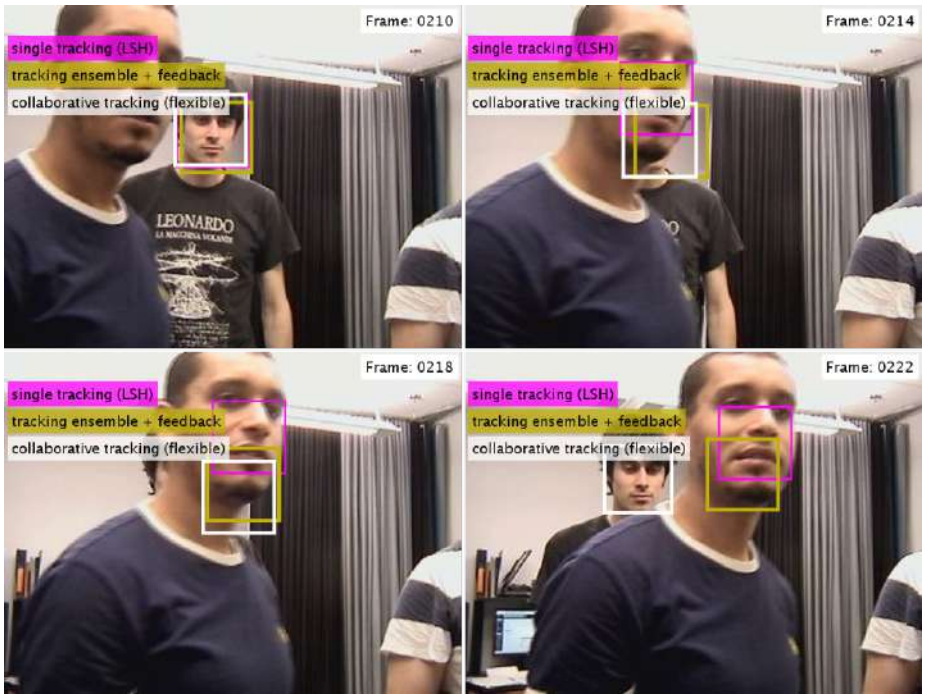
Figure 4.18: Framework's performance on multiple-face video sequences from the ChokePOINT Dataset. The graphics present the $AUC(e_{IoU})$ results obtained by combining trackers and complementing their fusion with a face detector; all of them for an intervention of the Inspector module at every $k = 1$, $k = 16$ and $k = 32$ frames, and a re-initialization buffer of size $s_b = 1$ (a) and $s_b = 32$ (b).

the minimum number of frames required for the trackers' to build a stable facial model, the facial states used in the trackers' re-initializations, or the quality of the image used in the trackers' re-initializations.

Figure 4.19 presents a subset of video frames of the MultiFaceFast and jp2 video sequences from the MOTINAS and LITIV datasets, respectively. The video frames contain the responses of the *collaborative tracking*(*flexible*) and *tracking ensemble + feedback* configurations, as well as the responses of the best performing tracker in the *single tracking* configuration with a continuous Inspection module intervention ($k = 1$). The figure shows that *collaborative tracking*(*flexible*) was the only configuration capable of recovering from a tracking failure caused by a severe occlusion over the target faces, which correspond to the individuals wearing black t-shirts.

In addition, the results suggest that the inclusion of the *cyclic dissimilarity buffer* does not offer considerable improvements in the face tracking using the *collaborative tracking* configuration. We noticed that the *volatile* variant remained inferior independent on the participation (different values of k) of the Inspection module in the framework prototype execution, whilst the *flexible* variant was the best performing one, getting closer to the *conservative* variant, as the ciclyc *dissimilarity buffer* size increases. The reason for this behavior is that by increasing the trackers' *cyclic dissimilarity buffer* sizes, the trackers

(a)



(b)

Figure 4.19: Video frames examples of the MultiFaceFast and jp2 video sequences from the MOTINAS (a) and LITIV (b) datasets. The figure illustrate the responses of the *collaborative tracking*(*flexible*) and *tracking ensemble + feedback* configurations, as well as the responses of the best performing tracker in the *single tracking* configuration with a continuous Inspection module intervention ($k = 1$).

became more conservative in re-initializing their facial models, which led to an update-variant-like behavior.

Moreover, when considering the constant execution of the Inspection module ($k = 1$), the *collaborative tracking* performed better than all of the other configurations, except for the ChokePOINT dataset. Following the analysis made in 4.5.1, we attribute it to the reference image used for the target face validation in the Inspection module. In fact, for the ChokePOINT dataset, the results corresponding to the *flexible* variant of the *collaborative tracking* configuration presented a curious increase in performance once the Inspection module intervention was restricted to every $k = 16$ frames. We relate this behavior to the high incorrect responses rates produced by the Inspection module, as discussed early on the validation experiment in Section 4.5.1. In our framework, an incorrect response from the Inspector module might lead a tracking to jump from the target face to another face, leading to an incorrect facial state estimate. Thus, as the Inspection module incorrect response rate decreases, the overall face tracking performance increases.

In this sense, we discuss, in the following, the results regarding the facial image used as a reference for the face target validation procedure in the Inspection module. We focus our discussion on the ChokePOINT dataset, which has already presented problems in respect to this cases, as already described in 4.5.1.

On the reference facial image for the target face validation

The three variants of the *collaborative tracking* configuration have shown better performances than the other tested configurations. The only exception was for the ChokePoint dataset, in which the best performing tracker from the *single tracking* configuration. We believe that this fact is a consequence of the target face validation procedure in the Inspection module, which primary task is to find the target face from the set of detection outcomes provided by the face detector. As already discussed in Section 4.5.1, the target face validation algorithm strongly depends on the facial image used as a reference. Moreover, its performance improves as the reference face approximates to a frontal posture, and decreases as it deviates from it.

Contrary to the tracking process in the TB-Face, LITIV and MOTINAS datasets, the tracking of some of the subjects in the ChokePOINT dataset started in frames where the target faces were far from having a frontal posture, which influenced in the overall tracking performance. Thus, to measure the impact of using facial images in nearly a frontal posture in the tracking process

within our framework, we conducted another experiment, which forces the framework to start the face tracking in frames where the faces of interest were in an almost frontal posture to the camera.

Similar to the original experiment on the ChokePOINT dataset, Figure 4.12 shows the performance results for the best tracker in the *single tracking* configuration, the fusion method from the *tracking ensemble + feedback* configuration and the three variants of the *collaborative tracking* configuration in the original experiment (indicated in black) and their analog variants for this new experiment (indicated in red): **volatile*, **conservative* and **flexible*.

Although the results in the new experiment corresponding to the **volatile*, **conservative* and **flexible* variants from the *collaborative tracking* configuration follow their respective original results tendency, there is a notable performance gain with respect to their original counterparts: *volatile*, *conservative* and *flexible*. Moreover, for the uninterrupted Inspection module operation ($k = 1$), the three variants outperformed the best tracker in the *single tracking* configuration. However, as its operation frequency decreases ($k = 16$ and $k = 32$), only the *flexible* variant remains superior to the best tracker in the *single tracking* configuration. These results reinforce our intuition that using a facial image in a nearly frontal posture as a reference for the target face validation procedure within the Inspection module actually helps on the framework performance and, hence, on the overall face tracking.

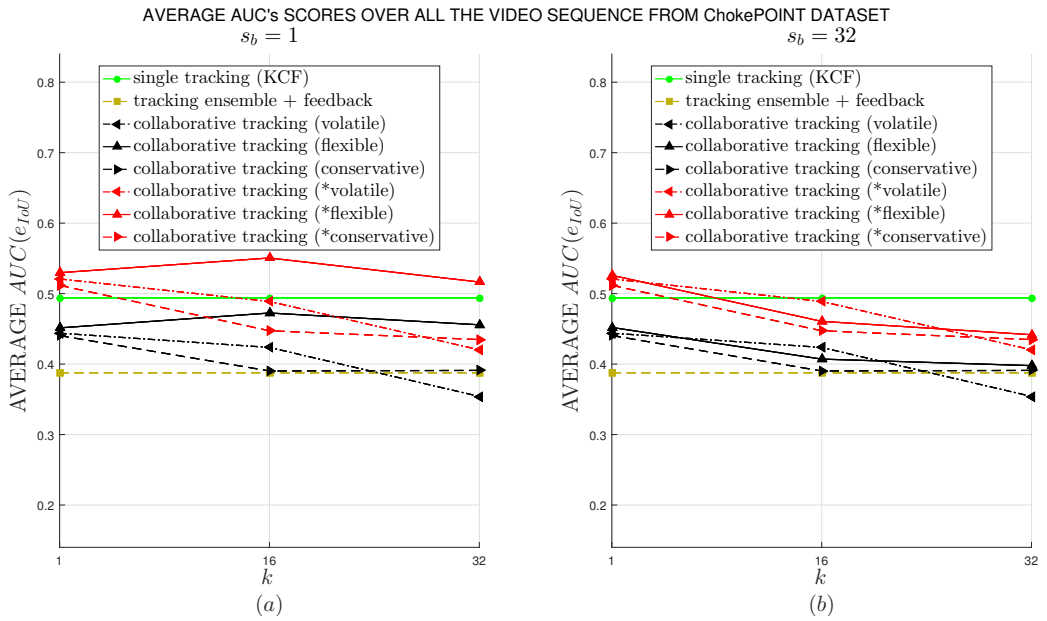


Figure 4.20: Framework's performance comparison on multiple-face video sequences from the ChokePOINT datasets using non-frontal and nearly frontal facial images as reference templates. The graphics present the $AUC(e_{IoU})$ results for an Inspection module intervention at every $k = 1$, $k = 16$ and $k = 32$ frames and a re-initialization buffer of size $s_b = 1$ (a) and $s_b = 32$ (b).

Finally, these results corroborate the inclusion of a face detector within the face tracking fusion offers significant gains in performance, specially in scenarios where the target face undergoes severe occlusions and has high resemblance among faces.

In this set of experiments, we have perceived that the best results correspond to the continuous operation of the Inspection module. However, comparable results can be obtained by allowing a regular operation frequency of this module at every 16 frames. Such behavior leaves space to explore new ways of enabling the face detector within the framework, for example, to activate the Inspection module only when one of the trackers begins to behave poorly or when the trackers diverge from each other.

5
Conclusions

In this research, we have presented a new framework for long-term face tracking, called Collaborative Face Tracking, which tends to be more robust to unconstrained imaging conditions and scenarios than individual trackers and ensembles, achieving superior performance by exploiting the cooperation among tracking algorithms, a tracking fusion process and a face detector, through a feedback learning mechanism.

The framework is organized into four modules: Tracking, Fusion, Inspection and Integration, which enables a cooperation scheme among trackers, tracking fusion algorithms and face detectors of any kind. Moreover, the modules' prime requirements are to accept and produce a rectangular bounding box as input and output, which allows a high flexibility degree.

The Tracking module admits any number of different tracking algorithms, despite their specific designs. To ensure a proper overall framework's execution, the trackers must only meet an operational processing chain regarding the target facial state prediction and correction, to later enable the facial model update or re-initialization. These modifications allow the trackers to work in an ensemble and to produce independent tracking estimates about the target face state.

The Fusion module allows the use of any tracking fusion algorithm that operates over the trackers' outputs from the Tracking module, which are represented only by rectangular bounding boxes. The resulting bounding box produced by the fusion procedure provides a more accurate estimation of the target face state.

The Inspection module allows the cooperation between any face detector and a target face validation procedure of any kind to provide a reliable facial state estimate of the target face. While the face detection algorithm is in charge of locating faces within the video frame being processed, the target face validation is used to find the target face from the pool of detection outcomes. The face validation procedure allows the framework to recover from tracking failures by adjusting the tracking trajectory, or recapturing the target face after a tracking loss. It is important to mention that the Inspection module may be executed in a given operation frequency, reducing the computational

costs.

Finally, the Integration module combines the Fusion and Inspection modules outcomes to provide the final tracking estimate. Furthermore, it uses this estimate to improve individual tracker performance through a feedback process to the Tracker Module. Then, the Tracker module measures the level of discrepancy between the final and a tracker estimate and send a command to update or re-initialize its facial model. A tracker's update involves adding new information about the target face into the tracker's facial model, whereas the re-initialization involves erasing its facial model and starting a new one from scratch based on the last processed frame.

We implemented the Collaborative Face Tracking in a software prototype using the C++ programming language. The prototype allows a batch mode execution of a set of video sequences. In order to validate the framework's prototype, we conducted a set of experiments upon facial video sequences corresponding to four facial tracking-specific datasets, namely, the TB-Face, LITIV, MOTINAS, and ChokePOINT datasets. The datasets offer challenging facial tracking conditions, emphasizing severe occlusions, in video sequences containing single and multiple faces. Furthermore, we produced reference annotations of the target faces for some of the subjects in the video sequences for the TB-Face, MOTINAS and ChokePOINT datasets, which can be reused by other researchers.

The experimental analysis demonstrated that the combination of multiple trackers reduces tracking drifts. Furthermore, the integration of the tracking process with a face detector substantially improves face tracking accuracy, specially in multiple-face scenarios, where the framework was able to recover from tracking failures and to recapture the target face after a tracking loss, severe occlusions, and subjects short time disappearances.

The results attested that combining the trackers' updating and re-initialization led to better performance than solely performing updating or re-initialization on every frame. In our experiments, the trackers were commanded to update or re-initialize based on their prior performance and a given re-initialization threshold δ_r. We also included a *cyclic dissimilarity buffer* to manage the update/re-initialization process. However, the results showed that collecting performance data of the trackers do not bring a significant difference in the overall framework's performance.

Throughout the experiments, we assessed the impact of executing the Inspection module every $k-$th video frames for different values of k. The results showed that the framework performed better when Inspection module always participated in the framework's execution, i.e., $k = 1$, which might not be

favorable for real-time applications. However, the results also suggest that it is possible to obtain similar results for higher values of k, which might represent a good balance between tracking accuracy and processing speed.

As stated in our experiments, the proposed framework outperformed different configurations like single trackers and ensembles, showing its benefits. However, the accuracy considering its absolute results is still not so high for some of the datasets, showing that these sequences are actually very challenging. In this sense, it is also important to note that the framework might be improved by simply adding new trackers, different fusion methods and different face detectors and validating algorithms.

Finally, although the Collaborative Face Tracking is focused to perform the long-term face tracking, its architecture enables the possibility of expanding the framework to another application fields. In theory, by replacing the face detector with a task-specific detector and the face-specific validation procedure with other general or task-specific purpose, the framework could be deployed in the tracking of pedestrians, cars, and so on.

5.1 Framework's Performance Remarks

The methods tested in our experiments were set to operate with their default parameter values, which might have prevented them to achieve their optimal performances. Investigating the best set of parameters of each method within the framework constitutes a separate work that goes beyond the scope of this work and can be investigated in future works.

Throughout the experiments, the Inspector module was set to collect a reference image of the target face at the beginning of the tracking, which is used to find the target face from the detection outcomes via a face validation procedure. A reference image having a facial image far from a frontal posture restrains the Inspection module capability to measure how similar a candidate and the reference templates are, making the Inspection module less effective.

5.2 Future Research

During the development of this research, we have identified some possible directions to continue with the improvement of this work in short and long terms.

We have designed our framework in a way that it allows the inclusion of any tracking algorithm. However, as this work evolved, we considered five conventional trackers to asses the performance of the framework: three

discriminative and two generative trackers of general purpose. So, it is natural to include more trackers of other natures to explore the framework performance in its full potential. In this respect, a possibility could be to include Neural Networks-based tracking techniques, for instance.

Furthermore, we expect that the differences in the trackers' designs yield to an improvement of the trackers' performances, and therefore to the overall framework's performance. So, we propose to evaluate the impact of executing the Tracking module with a different combination and number of trackers each time.

In this work, we have compared the framework's performance against those from the individual trackers', which is a good indicator of improvement. However, it would be interesting to include other tracking fusion algorithms for comparison in our analysis as an alternative to further validate our framework.

Regarding the target face validation procedure in the Inspection module, further research on the collection of the reference facial image could be carried out. For example, one option could be to collect facial images of the target face with each processed frame until one of the images presents a near frontal posture. Another option could be to use the first facial image to generate an alternative facial image that is frontal.

Another interesting field for research relates to the prototype's efficiency. The framework's structure allows investigating concurrent programming methods at different levels in the prototype for its faster execution. For instance, at an inter-module level, the Tracking and Fusion modules could be executed concurrently with the Inspector module to save time. At an intra-module level, the Tracking module, for example, offers many ways of concurrency: an alternative could be exploiting the trackers' execution per task within the processing chain described in Section 3.3. Finally, we also envisage a concurrency at a method level, which relates to the optimization of the methods, especially the tracking algorithms, using multiple processors or GPUs.

In particular, evaluating methodologies oriented to the tracking of faces is difficult as there is a lack of face-specific datasets with reliable reference annotations. In this research, we have collected a set of facial video sequences, furthermore we have annotated the references for some of the appearing faces. We expect to extend and make the annotations of the remaining faces in the video sequences available in the future.

Bibliography references

ASTHANA, A.; ZAFEIRIOU, S.; CHENG, S.; PANTIC, M. **Incremental face alignment in the wild**. In: IN PROCEEDINGS OF 2014 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2013), p. 1859–1866. IEEE, June 2014.

AYMA, V. H. Q.; HAPP, P. N.; COSTA, G. A. O. P. D.; FEITOSA, R. Q. **Symbiotic tracker ensemble with feedback learning**. In: PROCEEDINGS OF THE 2017 30TH SIBGRAPI CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES (SIBGRAPI), p. 421–428. IEEE, October 2017.

BABENKO, B.; YANG, MH.; BELONGIE, S. **Robust object tracking with online multiple instance learning**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(8):1619–1632, August 2011.

BAILER, C.; PAGANI, A.; STRICKER, D. **A superior tracking approach: Building a strong tracker through fusion**. In: PROCEEDINGS OF THE 13TH EUROPEAN CONFERENCE ON COMPUTER VISION - PART VII, Lecture Notes in Computer Science, p. 170–185, Cham, September 2014. Springer International Publishing.

BAKER, S.; MATTHEWS, I. **Lucas-kanade 20 years on: A unifying framework**. International Journal of Computer Vision, 56(3):221–255, February 2004.

BALLARD, D. H.; BROWN, C. M. **Computer Vision**. Prentice Hall Professional, Englewood Cliffs, NJ, USA, 1st edition, May 1982.

BERTINETTO, L.; VALMADRE, J.; HENRIQUES, J. F.; VEDALDI, A.; TORR, P. H. S. **Fully-convolutional siamese networks for object tracking**. In: COMPUTER VISION - ECCV 2016 WORKSHOPS, Lecture Notes in Computer Science, vol 9914, p. 850–865, cham, October 2016. Springer International Publishing.

BIRESAW, T.; CAVALLARO, A.; REGAZZONI, C. **Tracker-level fusion for robust bayesian visual tracking**. IEEE Transactions on Circuits and Systems for Video Technology, 25(5):776–789, May 2015.

BOUACHIR, W.; BILODEAU, GA. **Collaborative part-based tracking using salient local predictors**. Computer Vision and Image Understanding, 137:88–101, August 2015.

CHRYSOS, G. G.; ANTONAKOS, E.; SNAPE, P.; ASTHANA, A.; ZAFEIRIOU, S. **A comprehensive performance evaluation of deformable face tracking ""in-the-wild""**. International Journal of Computer Vision, 126(2):198–232, April 2018.

COLLINS, R. T.; LIU, Y.; LEORDEANU, M. **Online selection of discriminative tracking features**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10):1631–1643, October 2005.

COMANICIU, D.; RAMESH,V.; MEER, P. **Real-time tracking of non-rigid objects using mean shift**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 142–149. IEEE, June 2000.

COOTES T.; EDWARDS, G. J.; TAYLOR, C. J. **Active appearance models**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681–685, June 2001.

DANELLJAN, M.; ROBINSON, A.; SHAHBAZ KHAN, F.; FELSBERG, M. **Beyond correlation filters: Learning continuous convolution operators for visual tracking**. In: PROCEEDINGS OF THE EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV 2016), p. 472–488. Springer, October 2016.

FAN, H.; LING, H. **Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking**. In: PROCEEDINGS OF THE 2017 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (CVPR 2017), p. 5487–5495. IEEE, October 2017.

FORSYTH, D.; PONCE, J. **Computer Vision: A Modern Approach**. Prentice Hall, Upper Saddle River, NJ, 2nd edition, November 2011.

GAO, Y.; JI, R.; ZHANG, L.; HAUPTMANN, A. **Symbiotic tracker ensemble toward a unified tracking framework trackers-the "black boxes" approach**. IEEE Transactions on Circuits and Systems for Video Technology, 24(7):1122–1131, January 2014.

GRABNER, H.; GRABNER, M.; BISCHOF, H. **Real-time tracking via online boosting**. In: PROCEEDINGS OF THE BRITISH MACHINE VISION CONFERENCE, volumen 1, p. 47–56, September 2006.

HARE, S.; GOLODETZ, S.; SAFFARI, A.; VINEET, V.; CHENG, MM.; HICKS, S. L.; TORR, P. HS. **Struck: Structured output tracking with kernels**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10):2096–2109, October 2016.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep residual learning for image recognition**. In: IN PROCEEDINGS OF 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2016), p. 770–778. IEEE, June 2016.

HE, S.; LAU, R. W. H.; YANG, Q.; WANG, J.; YANG, MH. **Robust object tracking via locality sensitive histograms**. IEEE Transactions on Circuits and Systems for Video Technology, 27(5):1006–1017, May 2017.

HENRIQUES, J. F.; CASEIRO, R.; MARTINS, P.; BATISTA, J. **High-speed tracking with kernelized correlation filters**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(3):583–596, March 2015.

JAIN, R.; KASTURI, R.; SCHUNCK, B. G. **Machine vision**. McGraw-Hill New York, New York, NY, USA, August 1995.

JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; DARRELL, T. **Caffe: Convolutional architecture for fast feature embedding**. arXiv preprint arXiv:1408.5093, 2014.

JUN, B.; CHOI, I.; KIM, D. **Local transform features and hybridization for accurate face and human detection**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(6):1423–1436, June 2013.

KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. **Face-TLD: Tracking-Learning-Detection applied to faces**. In: PROCEEDINGS OF THE 2010 17TH IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, p. 3789–3792, September 2010.

KALAL, Z.; MIKOLAJCZYK, K; MATAS, J. **Tracking-learning-detection**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7):1409–1422, July 2012.

KRISTAN, M.; MATAS, J.; LEONARDIS, A.; VOJÍŘ, T.; PFLUGFELDER, R.; FERNÁNDEZ, G.; NEBEHAY, G.; PORIKLI, F.; ČEHOVIN, L. **A novel performance evaluation methodology for single-target trackers**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(11):2137–2155, November 2016.

LEANG, I. ; HERBIN, S.; GIRARD, B.; DROULEZ, J. **Robust fusion of trackers using online drift prediction**. In: ADVANCED CONCEPTS FOR INTELLIGENT VISION SYSTEMS, Lecture Notes in Computer Science, vol 9386, p. 229–240, Cham, October 2015. Springer International Publishing.

LEICHTER, I.; LINDENBAUM, M.; RIVLIN, E. **A general framework for combining visual trackers-the" black boxes" approach**. International Journal of Computer Vision, 67(3):343–363, May 2006.

LI, S. Z.; JAIN, A. K. **Handbook of Face Recognition**, chapter 1. Introduction, p. 1–11. Springer New York, New York, NY, USA, 1st edition, March 2005.

LI, J.; WANG, T.; ZHANG, Y. **Face detection using SURF cascade**. In: 2011 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (ICCV WORKSHOPS), p. 2183–2190. IEEE, November 2011.

LI, Q.; WANG, X.; WANG, W.; JIANG, Y.; ZHOU, ZH.; TU, Z. **Disagreement-based multi-system tracking**. In: COMPUTER VISION - ACCV 2012 WORKSHOPS (ACCV WORKSHOPS), Lecture Notes in Computer Science, vol 7729, p. 320–334. Springer Berlin Heidelberg, November 2012.

LI, H.; LIN, Z.; SHEN, X.; BRANDT, J.; HUA, G. **A convolutional neural network cascade for face detection**. In: 2015 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2015), p. 5325–5334. IEEE, June 2015.

LI, A.; LIN, M.; WU, Y.; YANG, M.; YAN, S. **Nus-pro: A new visual tracking challenge**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2):335–349, May 2016.

LI, P.; WANG, D.; WANG, L.; LU, H. **Deep visual tracking: Review and experimental comparison**. Pattern Recognition, 76:323–338, April 2018.

LIU, Y.; LI, H.; WANG, X. **Rethinking feature discrimination and polymerization for large-scale recognition**. CoRR, abs/1710.00870, 2017.

LUCAS, B. D.; KANADE, T. **An iterative image registration technique with an application to stereo vision**. In: PROCEEDINGS OF THE 7TH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, volumen 2, p. II–674–II–679 vol. 2. Morgan Kaufmann Publishers Inc., August 1981.

MA, C.; HUANG, JB; YANG, X.; YANG, MH. **Hierarchical convolutional features for visual tracking**. In: PROCEEDINGS OF THE 2015 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), p. 3074–3082. IEEE, December 2015.

MAGGIO, E.; PICCARDO, E; REGAZZONI C.; CAVALLARO, A. **Particle phd filtering for multi-target visual tracking**. In: PROCEEDINGS OF THE 2007 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP '07), volumen 1, p. I–1101–I–1104. IEEE, June 2007.

MAGGIO, E.; CAVALLARO, A. **Video Tracking: Theory and Practice**. Wiley, 1st edition, February 2011.

MATTHEWS, I.; ISHIKAWA, T.; BAKER, S. **The template update problem**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):810–815, June 2004.

MEI, X. AND LING, H. **Robust visual tracking using l1 minimization**. In: PROCEEDINGS OF THE 2009 IEEE 12TH INTERNATIONAL CONFERENCE ON COMPUTER VISION, p. 1436–1443. IEEE, September 2009.

RANJAN, R.; CASTILLO, C.; CHALLEPA, R. **L2-constrained softmax loss for discriminative face verification**. CoRR, abs/1703.09507, 2017.

RANJAN, R.; PATEL, V. M.; CHELLAPA, R. **Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition**. IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 1–1, 2017.

ROSS, D.; LIM, J.; LIN, RS.; YANG, MH. **Incremental learning for robust visual tracking**. International Journal of Computer Vision, 77(1):125–141, May 2008.

SHEARER, K.; WONG, K D.; VENKATESH, S. **Combining multiple tracking algorithms for improved general performance**. Pattern Recognition, 34(6):1257–1269, June 2001.

SIMONYAN, K.; ZISSERMAN, A. **Very deep convolutional networks for large-scale image recognition**. CoRR, abs/1409.1556, 2014.

SMEULDERS, A. WM; CHU, D. M.; CUCCHIARA, R.; CALDERARA, S.; DEHGHAN, A.; SHAH, M. **Visual tracking: An experimental survey**. IEEE

Transactions on Pattern Analysis and Machine Intelligence, 36(7):1442–1468, July 2014.

STENGER, B.; WOODLEY, T.; CIPOLLA, R. **Learning to track with multiple observers**. In: PROCEEDINGS OF THE 2009 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2009), p. 2647–2654, June 2009.

TAIGMAN, Y.; YANG, M.; RANZATO M.; WOLF. L. **DeepFace: Closing the gao to human-level performance in face verification**. In: IN PROCEEDINGS OF 2014 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2014), p. 1701–1708. IEEE, June 2014.

TAO, R.; GAVVES, E.; SMEULDERS, A. W. M. **Siamese instance search for tracking**. In: PROCEEDINGS OF THE 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), p. 1420–1429. IEEE, June 2016.

VIOLA, P.; JONES, M. **Rapid object detection using a boosted cascade of simple features**. In: PROCEEDINGS OF THE 2001 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2001), volumen 1, p. I–511–I–518 vol.1. IEEE, December 2001.

WANG, M.; DENG, W. **Deep face recognition: A survey**. CoRR, abs/1804.06655, 2018.

WONG, Y.; CHEN, S.; MAU, S.; SANDERSON, C.; LOVELL, B. C. **Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition**. In: IEEE BIOMETRICS WORKSHOP, COMPUTER VISION AND PATTERN RECOGNITION (CVPR) WORKSHOPS, p. 74–81. IEEE, June 2011.

WU, Y.; LIM, J.; YANG, M. **Online object tracking: A benchmark**. In: IN PROCEEDINGS OF 2013 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2013), p. 2411–2418. IEEE, June 2013.

WU, Y.; LIM, J.; YANG, MH. **Object tracking benchmark**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9):1834–1848, September 2015.

XIONG, X.; DE LA TORRE, F. **Supervised descent method and its applications to face alignment**. In: IN PROCEEDINGS OF 2013 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2013), p. 532–539. IEEE, June 2013.

YANG, MH.; KRIEGMAN, D. J.; AHUJA, N. **Detecting faces in images: A survey**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):34–58, January 2002.

YANG, H.; SHAO, L.; ZHENG, F.; WANG, L.; SONG, Z. **Recent advances and trends in visual tracking: A review**. Neurocomputing, 74(18):3823–3831, November 2011.

YILMAZ, A.; JAVED, O.; SHAH, M. **Object tracking: A survey**. ACM Computing Surveys, 38(4):13, December 2006.

ZAFEIRIOU, S.; ZHANG, C.; ZHANG, Z. **A survey on face detection in the wild: Past, present and future**. Computer Vision and Image Understanding, 138:1–24, September 2015.

ZHANG, C. AND ZHANG, Z. **A survey of recent advances in face detection**. Technical report, Microsoft Research, June 2010.

ZHANG, K.; ZHANG, L.; YANG, MH. **Fast compressive tracking**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(10):2002–2015, October 2014.

ZHANG, K.; ZHANG, Z.; LI, Z.; QIAO, Y. **Joint face detection and alignment using multitask cascaded convolutional networks**. IEEE Signal Processing Letters, 23(10):1499–1503, October 2016.

ZHONG, B.; YAO, H.; CHEN, S.; JI, R.; CHIN, TJ.; WANG, H. **Visual tracking via weakly supervised learning from multiple imperfect oracles**. Pattern Recognition, 47(2):1395–1410, March 2014.