Optimizing Process for Tracking People in video-camera network

Julien Badie

Université Nice Sophia Antipolis – UFR Sciences
École Doctorale STIC

THÈSE

Présentée pour obtenir le titre de :

Docteur en Sciences de l'Université Nice Sophia Antipolis

Spécialité : INFORMATIQUE

par

Julien BADIE

Équipe d'accueil : STARS – INRIA Sophia Antipolis

OPTIMIZING PROCESS FOR TRACKING PEOPLE IN VIDEO-CAMERA NETWORK

Thèse dirigée par François BRÉMOND

Soutenance à l'INRIA le 17 novembre 2015, à 10h00 devant le jury composé de :

Président :	Frédéric	PRECIOSO	Professeur, Université Nice Sophia Antipolis
Directeur :	François	BRÉMOND	DR2, INRIA Sophia Antipolis - Méditerranée
Rapporteurs :	Xavier	ROCA MARVA	Directeur du département des sciences informatiques, Barcelone
	Laure	TOUGNE	Professeur, Université Lyon 2
Examinateur :	Thierry	CHATEAU	Professeur, Université Blaise Pascal, Clermont-Ferrand

Thèse

Optimisation du suivi de personnes dans un réseau de caméras

Optimizing process for tracking people in video-camera network

Julien Badie

Novembre 2015

REMERCIEMENTS

Je veux tout d'abord remercier mon superviseur François Brémond pour m'avoir accepter comme doctorant dans son équipe PULSAR à l'époque devenu STARS aujourd'hui. Ces conseils et ses encouragements ont été très précieux. Il m'a toujours encouragé à suivre mes propres idées tout en me remettant dans dans la voie dans les moments d'égarement.

Je remercie Laure Tougne, responsable du LIRIS à Lyon et Xavier Roca Marva, directeur du département des sciences informatiques à Barcelone, pour avoir accepter d'être rapporteurs de ma thèse. Les remarques qu'ils ont faites ont été très pertinentes et m'ont permis d'améliorer ce manuscrit. Je remercie également Thierry Chateau pour son rôle d'examinateur et Frederic Precioso pour avoir accepter d'être président du jury.

Je remercie tous les membres de l'équipe STARS, les anciens comme les nouveaux, pour l'aide et le support qu'ils m'ont apportés tout au long de cette thèse. Je remercie tout particulièrement Duc Phu Chau et Slawomir Bak avec qui j'ai partagé mes travaux et qui m'ont également beaucoup appris. Je remercie également Julien Gueytat pour l'immense aide technique qu'il m'a apporté.

Je remercie chaleureusement Jane Desplanques pour sa disponibilité sans faille, sa précieuse aide pour les formalités administratives et son intarissable bonne humeur.

Je veux enfin remercier tous les membres de ma familles ainsi que mes amis, tout particulièrement Yacin, Frédéric et Julien qui m'ont encouragé et supporté durant cette thèse.

Abstract

Cette thèse s'intéresse à l'amélioration des performances du processus de suivi de personnes, dans un réseau de caméras et propose une nouvelle plate-forme appelée Global Tracker. Cette plate-forme évalue la qualité des trajectoires obtenues par un simple algorithme de suivi et permet de corriger les erreurs potentielles de cette première étape de suivi.

La première partie de ce Global Tracker estime la qualité des trajectoires à partir d'un modèle statistique analysant des distributions des caractéristiques de la cible (ie : l'objet suivi) telles que ses dimensions, sa vitesse, sa direction, afin de détecter de potentielles anomalies. Pour distinguer de véritables erreurs par rapport à des phénomènes optiques, nous analysons toutes les interactions entre l'objet suivi et tout son environnement incluant d'autres objets mobiles et les éléments du fond de la scène.

Dans la deuxième partie du Global Tracker, une méthode en post-traitement a été conçue pour associer les différentes tracklets (ie : segments de trajectoires fiables) correspondant à la même personne qui n'auraient pas été associées correctement par la première étape de suivi. L'algorithme d'association des tracklets choisit les caractéristiques d'apparence les plus saillantes et discriminantes afin de calculer une signature visuelle adaptée à chaque tracklet.

Finalement le Global Tracker est évalué à partir de plusieurs bases de données de benchmark qui reproduisent une large variété de situations réelles. A travers toutes ces expérimentations, les performances du Global Tracker sont équivalentes ou supérieures aux meilleurs algorithmes de suivi de l'état de l'art.

ABSTRACT

During the last years, significant improvements were made in detection, tracking and event recognition. More precisely, human event recognition has become an important research topic with many promising applications. However event recognition accuracy still relies a lot on detection and tracking quality. This thesis addresses the problem of improving the quality of the tracking results in a new framework, called Global Tracker, which evaluate the quality of people trajectory (obtained by simple tracker) and recover the potential errors from the previous stage.

The first part of this Global Tracker introduces a novel approach to estimate the quality of the tracking results without using any ground truth. This approach is based on a statistical model analyzing the distribution of the target features to detect potential anomalies. To differentiate real errors from natural phenomenon, we analyze all the interactions between the tracked object and its surroundings (other objects and background elements). Some methods to correct these errors directly or through feedback are also described.

In the second part, a new method is designed to improve the tracking quality and the general knowledge of the scene. This method tries to associate different segments of trajectory corresponding to the same person when the tracking algorithm fails to recover the person after long occlusions or when the person leaves the scene and re-enters. The first step is to select the most relevant appearance features to compute a visual signature for each segment of the trajectory. Then, an unsupervised learning step is used to perform the best matches between the segments.

Finally, this thesis proposes to evaluate each module separately and the whole Global Tracker in different scenarios and in association with different kinds of detection and tracking processes. Several datasets reproducing real-life situations are tested and the results are outperforming the state-of-the-art trackers.

CONTENTS

FIGURES

TABLES

1

INTRODUCTION

1.1 Motivation

In the past decades, processing power of computers and accuracy of sensors have greatly improved. Moreover, saving data has become easier and cheaper. Nowadays, many different applications rely partially or totally on large vision-based systems such as surveillance, biometrics or medical imaging. Even with the latest technological breakthroughs, it is very challenging to design a fully automatized system to analyze these data. Considering the enormous amount of data generated by these vision-based systems, a new problem has emerged: what are the optimal methods to process these data? Depending on the field of application, three different approaches are used: relying on one or several human operators (fig. 1.1), designing an autonomous system (fig. 1.2) or use an hybrid solution where the data are first pre-processed then analyzed by an operator.

Figure 1.1: Example of a vision-based system using human operators: a video surveillance control center (source: http://www.lgsinnovations.com/)

Figure 1.2: Example of an fully autonomous vision-based system: the prototype of the Google car

Designing a semi-autonomous or a full-autonomous system is a challenge that requires expertise from multiple different areas. In the case of systems based on video surveillance, the systems generally use the following processes(fig. 1.3) in order to achieve automatic understanding of the video content:

- A detection process. It detects every object moving and tries to categorize them (eg. people, vehicles, background objects, ...).
- A tracking process. It keeps track of the detected objects throughout the video stream.

- A high-level process. In the case of vision-based system dedicated to video surveillance, the main task of the high-level process is event recognition. The events can be very diverse from recognizing crowd behavior to detecting abnormal events such as people fighting or finding an abandoned luggage.

Figure 1.3: The three steps of a generic system based on video surveillance: detection, tracking and a high-level process

During the last years, significant improvements were made in detection, tracking and event recognition. However, most of the algorithms in the literature still make the hypothesis that the inputs given by the previous process are good enough to make their own algorithm work. As a matter of fact, these algorithms become very sensitive to errors because they cannot detect or manage them. In order to reduce the impact of input errors, some frameworks use a confidence score to measure the reliability of their inputs. That partially solves the problem as the algorithm will know which data can be used but it creates an high dependance between each elements inside the framework. This dependance can also be considered as beneficial as it allows better management of the data throughout the framework. Hence, a second flaw of State of The Art vision-based systems can be introduced: most of them are dedicated to one type of situation or one context.

Autonomous systems generally use learning methods and parameter tuning to improve their performance. The problem is that learning methods generally use training data from the same dataset which means that these training data have to be available and that unexpected events

will be even harder to retrieve due to the lack of training. Parameter tuning can also be tedious to optimize as it is mostly done manually.

The direct dependance between the input data and the framework used to process them can be explained by two concepts: the diversity of context and the diversity of scenario. The context is defined as all the elements that characterize a scene before starting the process (type and number of cameras, inside or outside zones, type of illumination, moving background). For a given context, many different kinds of scenario can happen that require a different type of processing. For example, analyzing a subway videos during a rush hour or the middle of the day are two very different scenarios that need a dedicated algorithm.

Considering these elements, our goal is to create a framework that improves the overall quality of tracking algorithms which is robust enough to context and scenario changes.

1.2 Objectives

The main objective of this thesis is to improve the overall quality of the tracking results by detecting and correcting errors made by the previous processing steps. In order to achieve this objective, we design a new framework called Global Tracker. The name 'Global Tracker' comes from the fact that this framework works as a post-tracking process and has knowledge of the entire scene and the previous steps of processing. The Global Tracker framework has three main tasks:

- Monitoring tracking outputs by estimating their quality without using ground-truth and by detecting anomalies

- Trying to correct some of these anomalies, either by repairing them directly or by sending signal to other modules

- Applying some high-level algorithms to improve tracking results that require predefined knowledge (such as trajectory smoothing or re-identification algorithms requiring previous tracking results)

1.3 Major contributions

The major contributions of this thesis are as follows:

1. The online evaluation method that performs a quality estimation of the tracking results. This evaluation is made during runtime by selecting key frames and extracting relevant and discriminating feature vectors to find potential anomalies.

2. Several methods to correct anomalies found by the online evaluation to improve the overall quality of the tracking results. These methods includes feedback to other algorithms, automatic parameter tuning and trajectory smoothing.

3. The re-acquisition and re-identification method for keeping track of people even if they disappear for a long time after an occlusion, after changing camera or after leaving and re-entering the scene. This method is also performed during runtime and is an adaptation of the re-identification problem, well-known in computer vision.

All these contributions are put together in a new robust framework called 'Global Tracker' described earlier.

1.4 Outline of the manuscript

This manuscript is divided into five chapters:

- Chapter I: Introduction. This chapter presents the motivation, objectives and main contributions of this thesis.

- Chapter II: Evaluation of tracker performance. This chapter describes the offline evaluation challenge of tracking algorithms and presents a new method to estimate the quality of tracking results during runtime.

- Chapter III: Tracklet matching over time. This chapter presents the re-acquisition and re-identification method to match people appearance in complex situations (even after a long occlusion or after leaving and reentering the scene).

- Chapter IV: Results and applications. This chapter shows the results of our Global Tracker framework compared to the state of the art and illustrates possible applications of this thesis with practical cases.
- Chapter V: Conclusion. This chapter summarizes the main contributions, results and presents future work and possible extensions of this thesis.

2

EVALUATION OF TRACKER PERFORMANCE

This chapter is an extended version of the publications:

- *"J. Badie, F. Brémond. Global tracker: an online evaluation framework to improve tracking quality. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2014."*

- *"D.-P. Chau, J. Badie, F. Brémond, M. Thonnat. Online Tracking Parameter Adaptation based on Evaluation. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2013."*

2.1 Introduction

Performance evaluation is crucial in estimating the quality of an algorithm on a given scenario or application. [Maggio & Cavallaro 2010] categorizes two kinds of method for performance evaluation: analytical methods and empirical methods (fig. 2.1). Analytical methods aim at evaluating a tracking algorithm from a theoretical point of view, taking into account complexity and requirements. On the other side, empirical methods evaluate directly the output of the tracking algorithm without considering how these results are computed. Empirical methods can also be divided in two categories depending on whether the ground truth is required or not to evaluate the results.

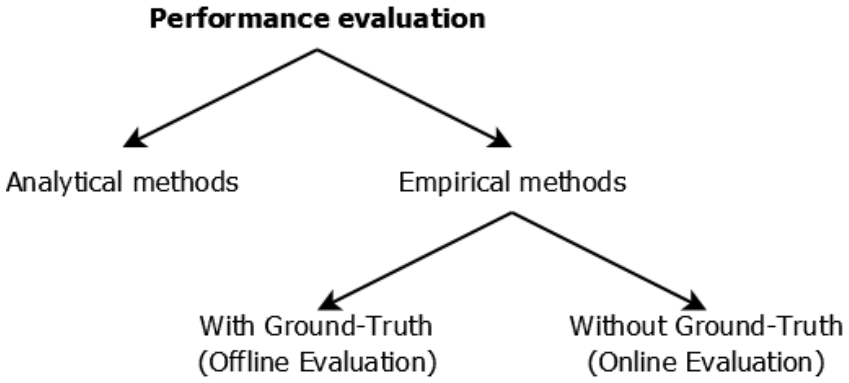


Figure 2.1: Different methods tracking performance evaluation

In this chapter, we first analyze the state of the art of empirical methods for evaluation using ground-truth. They are referred as *offline evaluation* methods because using ground-truth means the video has already been processed manually. Secondly, we present the first part of the Global Tracker, based on an evaluation method without ground-truth called *online evaluation*.

2.2 Offline evaluation

2.2.1 Related work

Offline evaluation is a common step to estimate the performance of a tracking algorithm. It is essential to estimate the results accuracy and it gives feedback to the user to find the optimal configuration for his/her framework. In the case of tracking performance evaluation, three inputs are necessary:

- The tracking results which have to be evaluated.
- The ground-truth which corresponds to the perfect results for detection and tracking with 100% accuracy
- One or several metrics which are distances for comparing the tracking results with the ground-truth.

However, even with fixed tracking results to evaluate, estimating the overall quality of an algorithm is still challenging because several ground-truths can exist for the same video sequence and because there exists a large variety of metrics, some more focused on evaluating the detection, some more focused on the tracking, some adapted to certain datasets and some more general. Some specific metrics can easily point out the strengths and weaknesses of an algorithm by looking at the situations that are correctly handled or not handled (for example occlusion management or shadow removal). However, the negative

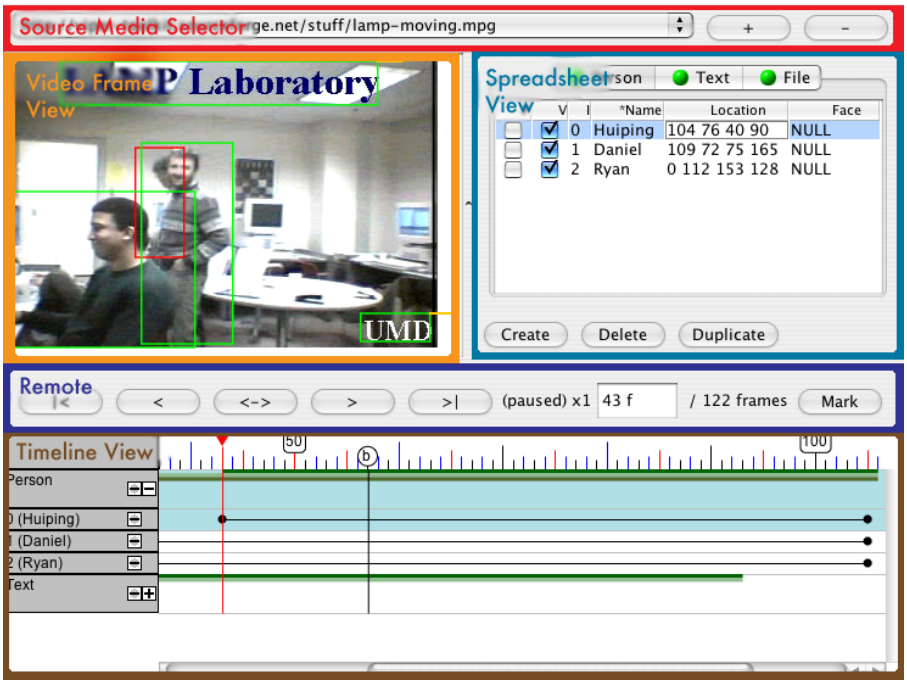
aspects are that it is harder and harder to compare the results with other algorithms from the state of the art and that the multiplicity of metrics can hardly give an overall impression on how well the algorithm really performs.

2.2.1.1 Ground-truth

In [Milan *et al.* 2013a], the authors make a first criticism of the non-uniqueness aspect of the ground-truth. For a same video sequence, there can be several ground-truths. The main reason for this is that ground-truths are generated using manual annotations and often using annotation tools. Specific methods can also help with annotation and labeling [Boom *et al.* 2012] [Boom *et al.* 2013] but remain generally dedicated to one type of data. Annotation tools such as ViPER [1] or VATIC [2] [Vondrick *et al.* 2013] propose an interface (fig. 2.2) to assist the user into the annotation process. However, since a human intervention is involved, the same sequence annotated by two different persons will most of the time result in two different ground-truths, even with the assistance of an annotation tool.

[1] http://viper-toolkit.sourceforge.net/

[2] http://web.mit.edu/vondrick/vatic/



(a) ViPER

(b) VATIC

Figure 2.2: GUI of two different annotation tools

The main source of ambiguity in annotation are occlusions. Generally, annotations are made by drawing bounding boxes around each target. However, it is up to the annotator whether to include occluded targets to the ground-truth. The decision is even harder with partially occluded targets, whether can be annotated with only their visible part, their full body or not annotated at all. The main challenge

of creating a ground-truth is whether it should be considered as a representation of the reality as we see it or as we imagine a vision-based system would see it. This difference of interpretation leads to the conclusion that ground-truth should not be considered as the most perfect results but is a good enough approximation of the optimal result. This also means that metrics should integrate this fact and not emphasize too much small errors coming from the difference of interpretation.

2.2.1.2 Metrics

A metric represents a distance between the tracking results and the ground-truth. Nowadays, many different sets of metrics exist and the most commonly used are VACE metrics [3] [Kasturi *et al.* 2009], CLEAR metrics [Bernardin & Stiefelhagen 2008], trajectory-based metrics [Wu & Nevatia 2006] and ETISEO metrics [Nghiem *et al.* 2007].

In order to describe the metrics, the following notations are used:

- G^i denotes the ith ground truth object in the sequence and G_t^i denotes the ith ground-truth object at frame t.
- D^i denotes the ith detected object in the sequence and D_t^i denotes the ith detected object at frame t.
- N_t^G and N_t^D denote the number of ground-truth objects and the number of detected objects at frame t, respectively.
- N^G and N^D denote the number of ground-truth objects and the number of detected objects in the sequence, respectively.
- N_{frames} is the number of frames in the sequence.
- N^{mapped} is the number of mapped ground truth and detected object pairs when the mapping is done at the sequence level and N_t^{mapped} is the number of mapped ground truth and detected object pairs at frame t (frame-level mapping).

VACE metrics

Sequence Frame Detection Accuracy (SFDA) The Sequence Frame Detection Accuracy (SFDA) evaluates the accuracy of the detection algorithm by first computing the Frame Detection Accuracy (FDA(t)). The Frame Detection Accuracy uses the overlap ratio between each detected object and the mapped ground truth object of the detected object, both represented as a bounding box at frame t.

[3] VACE - Video Analysis and Content Extraction, http://www.ic-arda.org

$$OverlapRatio = \sum_{t=1}^{N_t^{mapped}} \frac{|G_t^i \cap D_t^i|}{|G_t^i \cup D_t^i|} \tag{2.1}$$

The overlap ratio is then divided by the average number of detected objects and ground truth object on the frame to penalize the false positive and false detections.

$$FDA(t) = \frac{OverlapRatio}{\frac{N_t^G + N_t^D}{2}} \tag{2.2}$$

Finally, the Sequence Frame Detection Accuracy is computed as the average of the Frame Detection Accuracy over the sequence where at least one detected or ground truth object exists.

$$SFDA = \frac{\sum_{t=1}^{N_{frames}} FDA(t)}{\sum_{t=1}^{N_{frames}} \exists(N_t^G \vee N_t^D)} \tag{2.3}$$

Average Tracking Accuracy (ATA) The Average Tracking Accuracy (ATA) gives an overall appreciation of how well the tracking algorithm performs by first computing the Sequence Tracking Detection Accuracy (STDA) based on the overlap ratio between all the detections of a tracked object and all the detections of the mapped ground truth object. This metric assumes that there is a 1 to 1 match between the tracked objects and the ground truth objects.

$$STDA = \sum_{i=1}^{N^{mapped}} \frac{\sum_{t=1}^{N_{frames}} \frac{|G_t^i \cap D_t^i|}{|G_t^i \cup D_t^i|}}{N_{G^i \cup D^i \neq \emptyset}} \tag{2.4}$$

The Average Tracking Accuracy is then computed by dividing the STDA by the average number of tracked and ground truth objects.

$$ATA = \frac{STDA}{\frac{N^G + N^D}{2}} \tag{2.5}$$

CLEAR metrics

Multiple Object Detection Accuracy (MODA) The Multiple Object Detection Accuracy (MODA) gives another estimation of the detection accuracy by taking into account two types of tracking errors : the missed detections and the number of false positive, divided by the number of ground truth objects at frame t.

$$MODA(t) = 1 - \frac{m_t + fp_t}{N_t^G} \tag{2.6}$$

where m_t represents the number of missed detection count and fp_t represents the number of false positive.

Multiple Object Detection Precision (MODP) The Multiple Object Detection Precision (MODP) is defined as the overlap ratio divided by number of mapped ground truth and detected object pairs.

$$MODP(t) = \frac{OverlapRatio}{N_t^{mapped}} \tag{2.7}$$

Multiple Object Tracking Accuracy (MOTA) The Multiple Object Tracking Accuracy (MOTA) is an extension of the MODA on all the sequence taking into account still the number of false detections and false positive but also the number of ID switches between two detected objects.

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}}(m_t + fp_t + id_{switches})}{\sum_{t=1}^{N_{frames}} N_t^G} \tag{2.8}$$

Multiple Object Tracking Precision (MOTP) The Multiple Object Tracking Precision (MOTP) is an extension of the MODP on all the sequence.

$$MOTP = \frac{\sum_{i=1}^{N^{mapped}} \sum_{t=1}^{N_{frames}^t} \frac{|G_t^i \cap D_t^i|}{|G_t^i \cup D_t^i|}}{\sum_{t=1}^{N_{frames}} N_t^{mapped}} \tag{2.9}$$

Trajectory-based metrics

The trajectory-based metrics gives a global estimation of the tracking algorithm performance by computing the percentage of the trajectory of each object that corresponds to the ground truth. False alarms and ID switches are also part of these metrics:

- Number of ”Mostly Tracked” (MT) trajectories (more than 80% of the trajectory is tracked).
- Number of ”Partially Tracked” (PT) trajectories (between 20% and 80% of the trajectory is tracked).
- Number of ”Mostly Lost” (ML) trajectories (less than 20% of the trajectory is tracked).
- Number of false trajectories (false alarm/positive), (a result trajectory corresponding to no real object).
- The frequency of ID switches (identity exchanges between a pair of result trajectories).

ETISEO metrics

This set of metrics includes some already described metrics like the average overlap ratio between detected and ground truth objects.

Tracking Time The metric Tracking Time measures the percentage of time during which a reference data is detected and tracked. The match between a reference data and a detected object is done with respect to their bounding box. This metric gives us a global overview of the performance of tracking algorithms. It is similar to the trajectory-based metrics but keeps the raw results without classifying them.

Object ID Persistence The metric Object ID Persistence helps to evaluate the ID persistence. It computes over the time how many tracked objects are associated to one ground truth object. For example, a persistence of 1 means that a ground truth object is only represented by one tracked object even if the trajectory is not complete and there are some missing detections. A lower persistence means that the ground object is represented by more than one tracked objects which indicates that the tracking algorithm produces fragmented trajectories.

Object ID Confusion On the contrary, the metric Object ID Confusion computes the number of ground truth object IDs per detected object. A confusion of 1 still means that each ground truth object is associated with only one tracked object. However, a lower confusion means that at least one tracked object represents multiple ground truth objects which happens in the case of ID switches or when two IDs are incorrectly merged into one.

2.2.1.3 Metrics discussion

While many papers use the above-mentioned metrics, some others discuss the relevance of these metrics. Some of the metrics are indeed flawed and fail to give a correct representation of the tracking results.

The authors in[Milan *et al.* 2013a] highlight the fact that some metrics are ambiguous, especially the CLEAR metrics. More precisely, the metrics description [Kasturi *et al.* 2009] does not give all the information concerning the way to map a tracked object to a ground truth object. There is also another ambiguity concerning the computation of the metrics MODA and MOTA that involves three types of errors (missed detection, false positive and ID switch). The metric description suggests to use a weight functions associated to each type of error. However these functions are not clearly defined and so it is very likely that some of these metrics are not comparable due to the difference of weight functions.

Some of the VACE and CLEAR metrics also need to perform a matching between a tracked object and a ground truth object, which is a very common step in performance evaluation. However in the

detailed description of the metrics, it is stated that the matching is performed one to one, which means that in case of a trajectory interruption with a change of ID of the tracked object, the metrics will heavily penalize this.

The ETISEO metrics also have some flaws as the length of the tracked object trajectory does not interfere with the computation of the persistence and the confusion. For example, a ground truth object represented by two tracked objects covering the whole trajectory of the ground truth will have the same exact persistence than two tracked objects only covering half of the ground truth trajectory.

In conclusion, the metrics used nowadays are not perfect. The difference of implementation, mapping algorithm and also considering the fact that ground truth can vary make it hard for researcher to compare their results once and for all. Some other authors like [Smith *et al.* 2005], [Szczodrak *et al.* 2010], [Yin *et al.* 2007] and [Bernardin *et al.* 2006] try to bring new metrics, sometime more adapted to multi-object tracking. Finally, the best way to compare algorithms is in the case of tracking challenges like TRECVid or PETS, where the final evaluation process is made by the organization team. However, in this case, the evaluation protocols are most of the time not clearly detailed.

2.3 Online evaluation using contextual information

2.3.1 Tracklets

First, we need to formalize the output of the tracking algorithm. The output data of the tracking algorithm $\mathcal{O}$ can be represented as a list of tracklets. A tracklet is an oriented chain of nodes $\mathcal{C}^i$ representing one single object that appears on the scene with the ID i during the period $[T^i_{start}, T^i_{end}]$ (fig. 2.3). Each node $\mathcal{C}^i_t$ corresponds to one object detected at time t and contains all the data necessary for the tracking and post-tracking process. In most cases, the data are represented as features (eg. localization, appearance, ...). In the case of multi-cameras, an oriented chain can have multiple parallel sub-chains, one per camera.

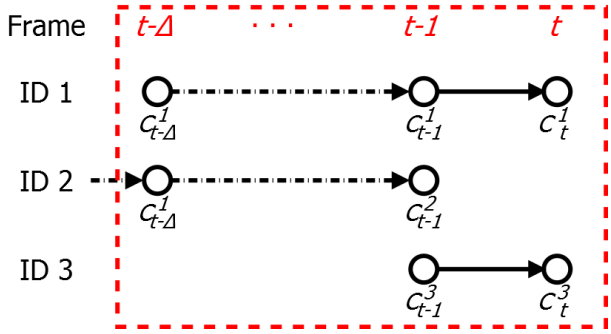


Figure 2.3: The tracklet representation within the time window as a chain of oriented nodes

All the processes described in this chapter and the following are defined on a sliding window of size Δ and have knowledge of everything that happened in this period. This knowledge is noted $\mathcal{K}_t$ and contains information on the interval $[t-\Delta, t]$. At time t, the goal of the process is to update the previous knowledge of the framework $\mathcal{K}_{t-1}$ using the output data of the tracking algorithm $\mathcal{O}_t$.

$$\mathcal{K}_t = f(\mathcal{O}_t, \mathcal{K}_{t-1}) \tag{2.10}$$

In the most favorable case, the knowledge pool $\mathcal{K}_t$ contains the following data:

- The current image
- All the previous tracking inputs $\mathcal{O}$ until current time t within the time window
- The outputs of the current process $\mathcal{O}'$ within the time window
- Features that are related to the scene $\mathcal{F}_t^{Scene}$ (eg. zones, known background elements, timestamp, ...)

This favorable case is when there is no constraint about memory space or processing time. Otherwise, the exact kind of data stored in this knowledge pool depends on the requirements of the whole process (online, offline, real-time, near real-time, ...).

$$\mathcal{K}_t = \begin{pmatrix} Image_t \\ \mathcal{O}_t \\ \mathcal{O}'_t \\ \mathcal{F}_t^{Scene} \end{pmatrix} \tag{2.11}$$

2.3.2 Hypotheses

In order to get optimal results with this online evaluation framework, we make some hypotheses on which cases it can be used and what type of information must be provided:

1. We cannot use any kind of ground truth. As stated in the introduction of this chapter, we aim to design a empirical method of evaluation that can be used while the system is running.

2. The vision-based system can be made of one or several overlapping or non-overlapping cameras but these cameras have to be fixed. As we mainly work with video surveillance systems or 3D cameras (for example the Kinect), this condition is nearly always met.

3. Inputs from the tracking algorithm are in majority correct. The main part of the online evaluation framework is to make a statistical analysis of the tracking data, so we need a minimum level of correct data to detect and correct the potential errors. We estimate that each segment of trajectory corresponding to one object need to be at least 50% correct for the online evaluation to work. This percentage may vary depending on the type of error encountered.

4. Prior knowledge about the scene is not required because we want to keep the online evaluation framework as generic as possible but these information can be used if available to increase the robustness of the process.

5. The process does not have to be real-time but outputs have to be processed fast enough to be followed by a human operator.

2.3.3 Interpolation

Some frames may be missing in an object trajectory. It happens if the tracking algorithm fails to find a correct match for an object on the current frame but is nevertheless able to recover the trajectory on the next frame. According to the tracklet representation, it means that some nodes are missing in the tracklet. The assumption is made that the tracking algorithm can not create tracklets with more than five consecutive nodes missing. If this case were to happen, the object would be considered as lost and its ID would not be used anymore, meaning that the chain has ended.

In order to fill the missing nodes, linear interpolation is performed using the feature pools of the two nodes located just before and just after the missing nodes 2.4.

$$\exists t \in [T^i_{start}, T^i_{end}] : \mathcal{F}^i_t = \emptyset \Rightarrow \mathcal{F}^i_t = \frac{\mathcal{F}^i_{t-1} + \mathcal{F}^i_{t+1}}{2} \tag{2.12}$$

where $\mathcal{F}^i_{t-1} \neq \emptyset$ and $\mathcal{F}^i_{t+1} \neq \emptyset$

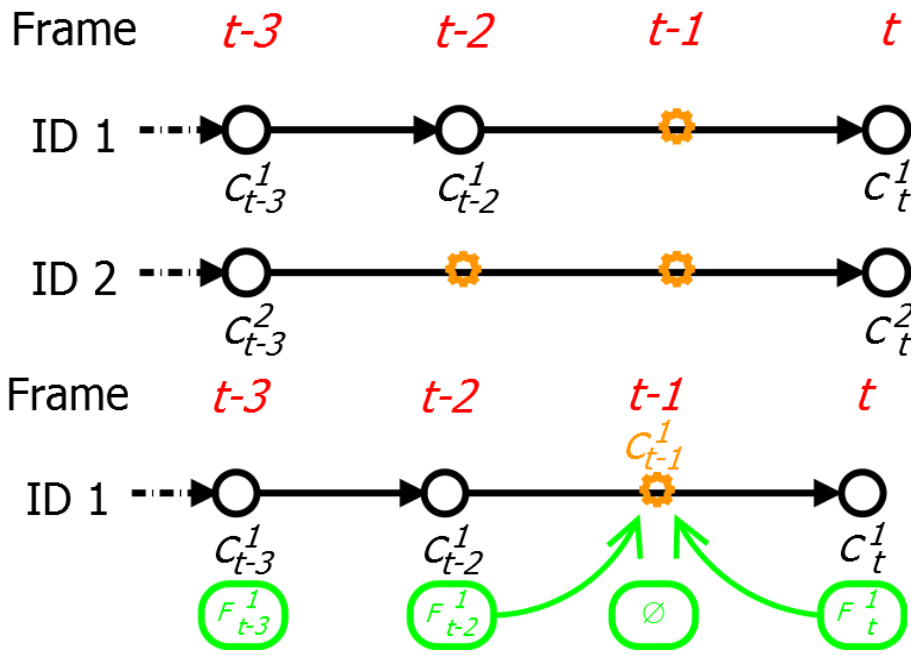


Figure 2.4: Interpolation considering the graph representation of a tracklet

In the case where several consecutive nodes are missing, the same interpolation method is used with the last two known nodes. Due to the assumption that a new tracklet is created if more than five consecutive nodes are missing, there is no need to use a more elaborated and time-consuming method to fill the missing nodes. Considering this assumption and the fact that the interpolation module is used at every frame, it can be sure that each tracklet contains no empty nodes. The main goal of this interpolation process is to optimize the tracking input for the upcoming process. It also slightly improves the tracking results as the interpolation constitutes a first step to a trajectory smoothing algorithm.

2.3.4 Anomalies detection and classification

Our goal is to design an online evaluation framework. Without any help from a ground truth, one solution is to check how each tracklet evolves over time and detect if there are some abnormal nodes within the tracklet.

Each node of each tracklet is characterized by various features. These features form a set of characteristic elements called a feature pool. The feature pool $\mathcal{F}_t^i$ of each node $\mathcal{C}_t^i$ is divided into three feature pools $\mathcal{F}_t^i = \{\mathcal{F}_t^{O,i}, \mathcal{F}_t^{OO,i}, \mathcal{F}_t^{OE,i}\}$:

- $\mathcal{F}_t^{O,i}$ represents the pool of features that are computed only using the data of the object (e.g. appearance, trajectory, ...)

- $\mathcal{F}_t^{OO,i}$ represents the pool of features that are computed using data of the object i considering the other objects of the scene (eg. occlusion level, people density, ...)
- $\mathcal{F}_t^{OE,i}$ represents the pool of features that are computed using data of the object i considering the environment (eg. occlusion level with background element, entering or leaving some zones, ...)

As we aim to detect anomalies inside the tracklet, we try to find nodes that contains abnormal feature values compared to the rest of the tracklet. In order to monitor the tracklet evolution, the feature pool $\mathcal{F}^{O,i}$ is the most relevant. The other feature pools $\mathcal{F}^{OO,i}$ and $\mathcal{F}^{OE,i}$ are used to define the neighborhood of the object and will help to classify the anomalies found. An other fact to be considered is that in an online process, we want to compare the latest node with the previous ones. In this case, the feature values of the object alone should be closer to the feature values of the most recent nodes rather than to the first nodes of the tracklet. This is reason why, for each feature $f^i \in \mathcal{F}^{O,i}$, we compute the weighted mean $\mu(f^i)$ and the weighted standard deviation $\sigma(f^i)$.

$$\mu(f^i) = \frac{\sum_{t=T^i_{start}}^{T^i_{end}} w(t) * f^i_t}{\sum_{t=T^i_{start}}^{T^i_{end}} w(t)} \tag{2.13}$$

$$\sigma(f^i) = \sqrt{\frac{\sum_{t=T^i_{start}}^{T^i_{end}} w(t) * (f^i_t - \mu(f^i))^2}{\sum_{t=T^i_{start}}^{T^i_{end}} w(t)}} \tag{2.14}$$

where w is the weight function and $[T^i_{start}, T^i_{end}]$ is the time interval where the tracklet is defined. This weight function is used to decrease the impact of the oldest nodes while focusing on the latest nodes. For our experiment, two different weight functions are tested, a linear function and an exponential function. The exponential function generally gives better results when heavy changes occur in the tracklet (mainly change of pose). Finally, the coefficient of variation $c(f^i)$ of each feature is computed.

$$c(f^i) = \frac{\sigma(f^i)}{\mu(f^i)} \tag{2.15}$$

A sudden and important change of feature values may indicate potential errors whereas keeping the same values should normally indicate that the tracking is correctly performed. Taking this into consideration, we define $\delta(f^i)$, the function that compares the coefficient of variation with the latest frame at t with the previous coefficient of variation at frame $t-1$.

$$\delta(f^i) = \left| 1 - \frac{c(f^i)_t}{c(f^i)_{t-1}} \right| \tag{2.16}$$

If δ^i is near or equal to zero, it means that the last node of the tracklet has the same behavior as the previous nodes of the tracklet. Otherwise, it means that the last node is diverging from the rest of the

tracklet. In that case, an anomaly is detected. For the experiments, a threshold of 0.25 is set to know if the value of δ could indicate an anomaly. Moreover, δ is a function related to only one feature and we consider that having one value of δ below the threshold is enough to detect an anomaly.

The next step is to determine whether the anomalies found are real errors or just natural phenomena. For example, an anomaly is found when a tracked object becomes occluded by a background element of the scene (the size of the bounding box and appearance of the object suddenly change) and this kind of anomaly should be categorized as a normal phenomenon if the object is disappearing but as an error if the bounding box of the object is merged with the background element. That is why the feature pools $\mathcal{F}_t^{OO,i}$ and $\mathcal{F}_t^{OE,i}$ are used. These feature pools contain information about the neighborhood of the object. Depending on the used features, contextual information such as entering or leaving zones in the scene can easily discriminate normal phenomena from real errors. Section 2.3.5 defines all the pools of features used for the experiment and their influence on the anomaly categorization.

In this chapter, we use a basic error correction system based on removing the faulty nodes and using contextual information to either wait for the next frame to restore this new missing node by interpolation or keep the node empty for example in the case of wrong detection. The chapter 3 show more complex methods to correct the errors found and chapter 4 give the final results with the online evaluation framework included into the Global Tracker.

2.3.5 Feature used for online evaluation

As said before, the feature pool of each node is divided into three feature pools $\{\mathcal{F}_t^{O,i}, \mathcal{F}_t^{OO,i}, \mathcal{F}_t^{OE,i}\}$ depending on which elements are used to compute the feature: the tracked object alone, the tracked object and other objects or the tracked object and the environment. In this section, we describe which features are used for the experiments. We choose to use the same set of features for all the datasets to keep the online evaluation as generic as possible. Choosing a more dedicated set of features depending on the video sequence is a problem that is not addressed in this thesis. Table 2.1 sums up the features used for the experiments.

Tracked object features

Bounding box dimensions: The width and height of the tracked object are computed. This feature is sensitive to large variation in the object detection, which can be symptomatic of a merge or a split between different objects.

Trajectory: The direction and the speed are computed. This feature is sensitive to sudden changes in

the path of the tracked object, which could indicate a swap between the IDs of two objects for example.

Color histogram: RGB color histograms are computed. This feature can be used to detect big changes with the tracked object appearance that can occur because of a false detection or a ghost phenomenon.

Covariance-based appearance model [Bak *et al.* 2011]: This feature is an alternative to the color histogram feature. It is more accurate than color histogram and has shown very good results in the re-identification domain. This is the main feature used for the re-acquisition and re-identification module. For each object, a visual signature is computed and updated at every frame. If a significant change is detected by computing the distance between the initial signature and the updated one, this can be interpreted as a possible error in detection or tracking.

Features characterizing the tracked objects versus other tracked object

Density with other objects: This feature is used to estimate the possible interactions between two objects at the same time t. If two objects are very close to each other it can be the origin of a detection or tracking error. Density is computed as follows considering two detected objects C_t^1 and C_t^2:

$$density = \frac{Area(C_t^1) + Area(C_t^2)}{Area(C_t^1 \cup C_t^2)} \tag{2.17}$$

Spatial overlap level with other objects: This feature computes the spatial overlap between all tracked objects that are overlapping (non-zero intersection of the bounding boxes) at the same time t. It is defined by the maximum ratio between the intersection of both bounding boxes and the bounding box that has the biggest area:

$$spatialOverlap = max\left(\frac{C_t^1 \cap C_t^2}{Area(C_t^1)}, \frac{C_t^1 \cap C_t^2}{Area(C_t^2)}\right) \tag{2.18}$$

Frame-to-frame overlap with other objects: This feature works as the spatial overlap feature except it is computed with one object C^1 at frame t and all other objects with a different ID at frame $t-1$. If the intersection exists with at least one other object C_{t-1}^2, the frame-to-frame overlap is:

$$f2fOverlap = \frac{C_t^1 \cap C_{t-1}^2}{Area(C_t^1)} \tag{2.19}$$

Features characterizing the tracked objects versus the environment

The following features are part of the object versus environment feature pool $\mathcal{F}_t^{OE,i}$. They can be computed depending on the scene meta-data and are directly related to the fourth hypothesis made in

section 2.3.2 stating that this kind of information is not required but can improve the accuracy of the online evaluation.

In the case where scene information are available, some zones can be defined (background elements, zones where people can enter/leave). If this step cannot be performed, the zones can be learned online using a statistical method on a part of the sequence, considering as an initial situation, that the borders of the image are zones where people can enter and leave. The zones are then updated each time a tracked object appear or disappear. In the results part, only the offline method is evaluated because the online method to detect zones requires long video sequences (more than one hour) with a sufficient number of people to be effective.

Object appearing/disappearing in zone: When a detected object with a new ID appears or disappears (new tracklet), we check if this happens in a zone where the object can leave/enter the scene. If this happens outside of these zones, an anomaly is detected.

Spatial overlap level with background elements: Depending on the scene meta-data, it is possible to know if the object is being occluded by a background element.

Feature pool	Feature description
$\mathcal{F}^{O}$	bounding box dimension
	trajectory (direction + speed)
	color histogram
	covariance matrices
$\mathcal{F}^{OO}$	density with other objects
	spatial overlap level with other objects
	frame-to-frame overlap with other objects
$\mathcal{F}^{OE}$	object appearing/disappearing in zone
	overlap level with background elements

Table 2.1: Features used for experimentation

2.4 Results

The evaluation process of this first module of the Global Tracker is achieved by two different series of test. The first tests aims to assess if the errors found by the online evaluation process are real errors in regards to the ground-truth. The second tests compare the tracking results of the proposed approach with the State of The Art on two different datasets.

2.4.1 Error detection accuracy

The methodology to compare errors found by the proposed approach and errors given by the ground-truth is described in figure 2.5.

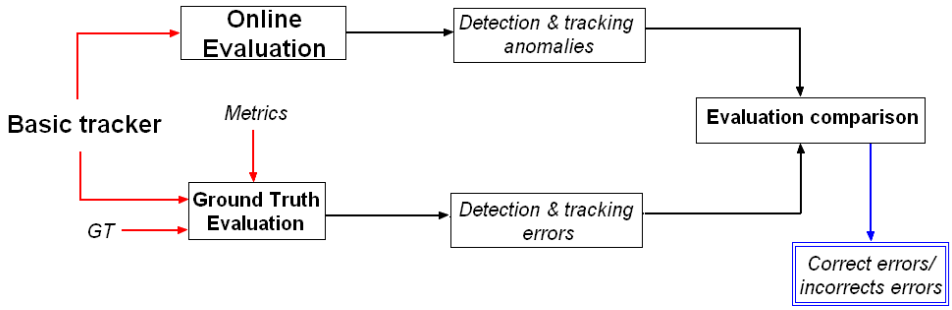


Figure 2.5: The methodology to evaluate the proposed approach is based on the comparison with the ground-truth

We use the dataset PETS2009 to evaluate the accuracy of our error detection process. The sequence S2.L1 is particularly interesting because the tracking algorithm is likely to encounter many types of different errors (people no detected, challenging occlusions, people merging/splitting). To experiment the proposed approach, we use a detection algorithm based on standard background subtraction. For tracking, two different algorithms are tested. One is a multi-feature tracker (**Tracker 1**) that uses 3D position, shape, dominant color and HOG descriptors. The other one is an algorithm based on graph partitioning (**Tracker 2**). Table 2.2 shows the percentage of correctly found errors (true positive), incorrectly labeled errors (false positive) and not found errors (false negative). .

Tracker	Errors from GT	True positive (%)	False Positive (%)	False Negative (%)
Tracker 1	306	65.60%	6.86%	34.4%
Tracker 2	165	60.61%	9.09%	39.39%

Table 2.2: Percentage of errors found the online evaluation compared to the errors given by the ground-truth for the sequence S2.L1 of PETS2009

Overall, the online evaluation process is able to detect more than 60% of the tracking error. The exact tracking performance of the two trackers with and without the online evaluation can be found in the next section. The most easily detected type of errors are errors due to occlusions 2.6 and group of people merging. However the online evaluation is unable to detect the missing detections of a person in the background or when the error can be found since the beginning of the tracklet 2.7.

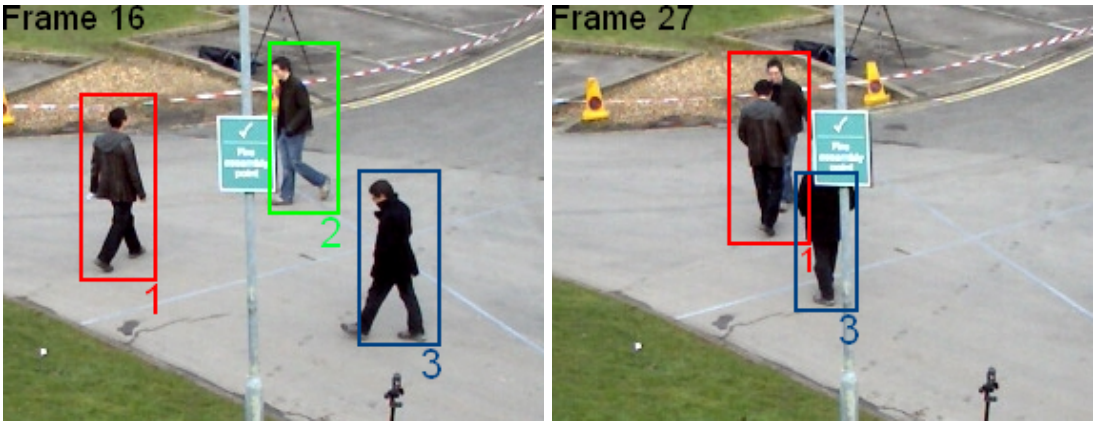


Figure 2.6: Easily found error due to occlusion

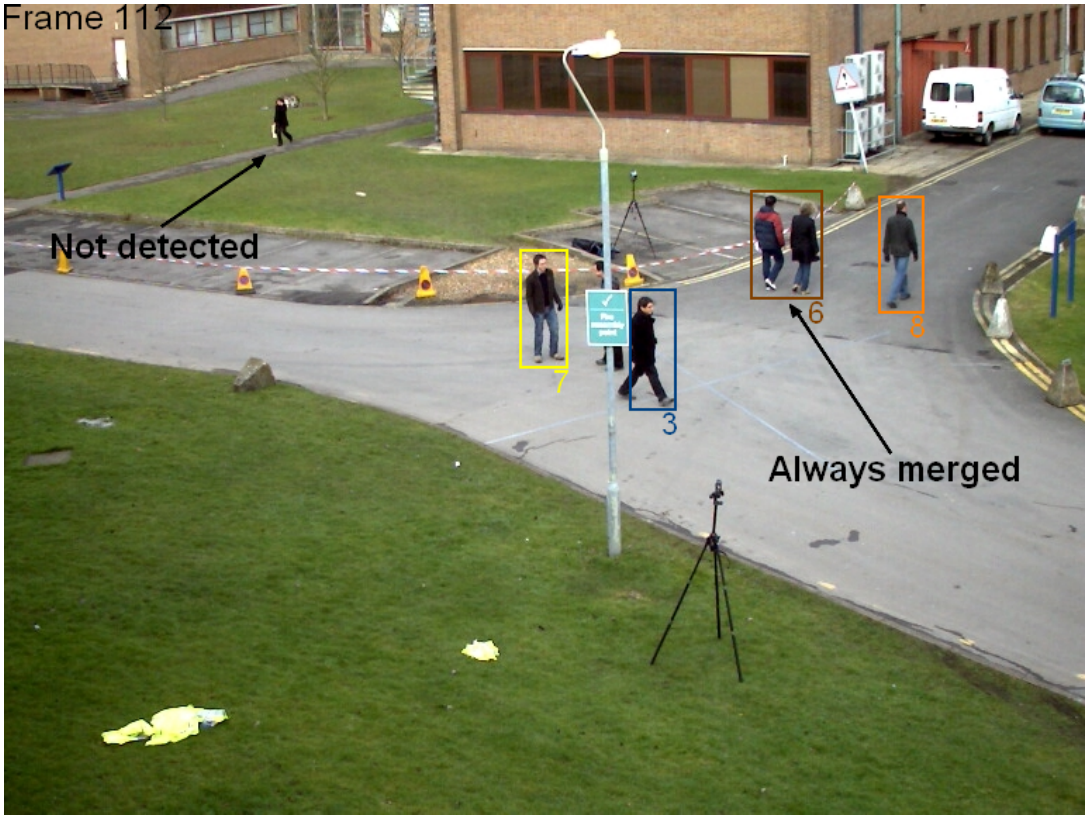


Figure 2.7: Limits of the online evaluation : missing detections and continuous errors cannot be detected

The figure 2.8 shows the comparison between the percentage of errors found by the online evaluation and the percentage of ground truth errors found over time in the tracking output for sequence S2.L1.View1 of PETS2009. For most errors, the online evaluation is able to successfully detect the errors (both curves match). However it fails at the beginning of the video (around frame 75) because the online evaluation has difficulties to detect errors from a tracklet that has a consistent error since its entrance in the scene.

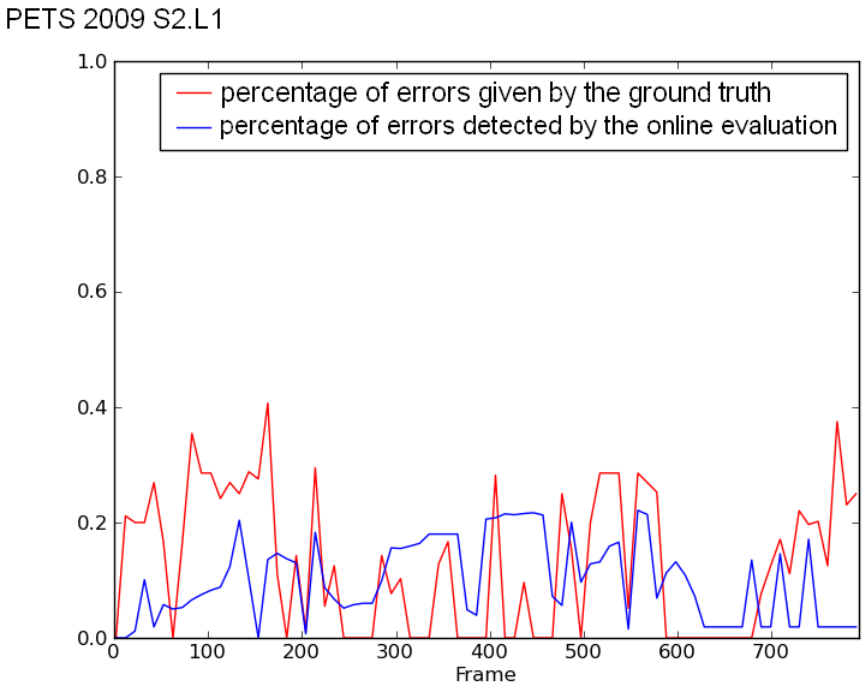


Figure 2.8: Comparison between the percentage of errors per frame given by the ground truth (red) and the percentage of errors found by the online evaluation (blue) on sequence S2.L1.View1 of PETS2009 using tracker 1

2.4.2 Tracking results with the online evaluation process

The online evaluation framework is evaluated on the public datasets PETS2009 and Caviar to compare the tracking results with the State of The Art. The detailed description of these datasets can be found in chapter 4. The same setup described in the last section with the two trackers (**Tracker 1**) and (**Tracker 2**) is used.

For the PETS sequence S2.L1, we use the CLEAR MOT metrics in order to compare with the state of the art. The metric MOTA is Multiple Object Tracking Accuracy which measures the number of false positives, false negatives and ID switch. The metric MOTP is Multiple Object Tracking Precision which measures the alignment of the tracking results with the ground truth.

Table 2.3 gives the results of the tracking algorithm with and without the online evaluation for the two trackers.

Methods	MOTA	MOTP
[Berclaz *et al.* 2011]	0.80	0.58
[Ben Shitrit *et al.* 2011]	0.81	0.58
[Henriques *et al.* 2011]	0.85	0.69
[Zamir & Shah 2012]	**0.90**	0.69
[Milan *et al.* 2013b]	**0.90**	**0.74**
Tracker 1	0.62	0.63
Tracker 1 + online evaluation	0.85	0.71
Tracker 2	0.85	**0.74**
Tracker 2 + online evaluation	**0.90**	**0.74**

Table 2.3: Tracking precision and accuracy on sequence S2.L1.View1 of the PETS2009 dataset with and without using the online evaluation framework

The results show that while tracker 1 has difficulties providing reliable output results, the online evaluation is able to improve the quality of the results by increasing the MOTA from 0.62 to 0.85 and the MOTP from 0.63 to 0.71. In the case of a tracker that already gives satisfying results (tracker 2), the online evaluation is able to make the results even better by increasing the MOTA from 0.85 to 0.90 while keeping the same value for the MOTP (0.74).

The online evaluation is also evaluated on the Caviar dataset. For this dataset, another set of metrics is used according to the state of the art: Mostly Tracked (MT) means that more than 80% of the trajectory is correctly tracked, Partially Tracked (PT) means that between 20% and 80% of the trajectory is correctly tracked and Mostly Lost (PT) means that less than 20% of the trajectory is correctly tracked. Table 2.4 gives the results of the tracking algorithm with and without online evaluation for trackers 1.

Method	MT (%)	PT (%)	ML (%)
[Xing *et al.* 2009]	84.3	12.1	3.6
[Huang *et al.* 2008]	78.3	14.7	7
[Li *et al.* 2009]	84.6	14.0	1.4
[Kuo *et al.* 2010]	84.6	14.7	**0.7**
Tracker 1	78.3	16.0	5.7
Tracker 1 + online evaluation	82.6	11.7	5.7

Table 2.4: Tracking results on the Caviar dataset

The results show that the online evaluation is able to improve the tracking results by increasing the length and the precision of the tracklets. The main drawback is that it fails to correct the tracklets that are already mostly lost due to the lack of correct input detection.

2.5 Conclusions

In this chapter, we present a new framework for evaluating the quality of people trajectories during runtime. The challenge of online evaluation is to be able not to use ground-truth (a priori knowledge on expected results) but still be able to conduct a meaningful evaluation. To assess the quality of the tracking output, we analyze the distribution of tracked object features over the time and when significant changes or incoherence features are detected, these changes are classified into processing errors or natural phenomena. Natural phenomena correspond to well modeled feature variations such as a person is leaving the scene from an exit zone or a person is occluded by a registered background element. This is the first module of the Global Tracker framework. With complex features, the processing time of the online evaluation is around 2 seconds to detect error for tracklets over 10 frames, leading to an average of 5 frames per second corresponding to a reasonable frame rate. The online evaluation is assessed on two datasets and shows promising results by improving the tracking performance thanks to a simple recovering process used in post-treatment when an error is detected. Another experimentation shows the high correlation between the predicted errors and errors which have been detected using ground-truth in an offline evaluation process.

Several extensions of this work can be performed. The next step for this online evaluation framework is to select and choose automatically the more adequate set of features. A feedback stage from the online evaluation process towards the tracking algorithm could also be designed in order to provide another

mechanism to correct the detected errors. The obtained tracking algorithm could be then considered as a self-tuning algorithm. Another possible extension of this work would be to run in parallel the online evaluation on several tracking algorithms and select the best one according to their performance on the current video stream. This mechanism extends the notion of particle filters for classical people tracking algorithm.

3

TRACKLET MATCHING OVER TIME

This chapter is an extended version of the publications:

- *"J. Badie, S. Bak, S.-T. Serban, F. Brémond. Recovering people tracking errors using enhanced covariance-based signatures. In PETS 2012 workshop, associated with IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2012."*
- *"S. Bak, D.-P. Chau, J. Badie, E. Corvee, F. Brémond, M. Thonnat. Multi-target Tracking by discriminative analysis on Riemannian Manifold. In IEEE International Conference on Image Processing, 2012."*

3.1 Introduction

Recognizing and memorizing people is a usual and unconscious task for human brain. Even if we have some clues on how this recognition is performed [Sinha 2006], it is impossible for a computer to replicate the human brain process. In video analysis, human recognition is still a challenge that can be addressed at three different levels:

- Human recognition on a single frame, commonly called people detection.
- Human recognition on consecutive frames, commonly called people tracking.
- Long-term human recognition. It is used to determine whether a given person has already been observed in a network of cameras and is commonly referred as human re-identification.

This last challenge is newer compared to the others and is particularity relevant to the video surveillance domain.

In this chapter, we present the second part of the Global Tracker which focuses on addressing the long-term human recognition challenge by adapting a method already used in re-identification. In order to achieve this goal, we select key frames that give the most relevant representation of each tracklet and we compute a highly discriminative visual signature based on these key frames. The problem of matching several tracklets belonging to the same individual is handled as a ranking problem using unsupervised learning. The final objective is to group all the tracklets corresponding to the same individual.

3.2 Related work

In the state of the art, many different appearance-based descriptors with satisfying results already exist. A global tracking approach is described in [Chau *et al.* 2009] to correct lost trajectories thanks to learned scene semantic information. In [Wu & Nevatia 2007], a tracking method based on body parts is proposed using edgelet features. In [Kuo *et al.* 2010], the authors present a reliable descriptor and tracklet association method. However, in the majority of cases, two problems remains unsolved: the discrimination of the visual signatures (except in [Kuo *et al.* 2010]) and the size of the time window where the algorithm can fuse two trajectories. To summarize, the state of the art has focused more on repairing trajectory interruptions due to short-term occlusions than checking if a person has left and reentered the scene.

3.3 Re-acquisition and re-identification method

3.3.1 Re-acquisition/re-identification versus classical re-identification

Nowadays, re-identification has become a challenging task for high-level recognition. It can be defined as calculating whether a given person has already been observed in a network of cameras. As we are mainly interested in video surveillance applications and as part of the Global Tracker framework, we can ignore the methods based on iris, face, gait or silhouette because the precision of the sensors does not allow to extract this kind of visual signature. However, it is possible to rely on people appearance and this kind of challenge is known as appearance-based re-identification.

Appearance-based re-identification challenge is often introduce as follows:

1. There is one query that can be a photo of a person or a tracklet representing that person. The task is to find all the time and space occurrences of that person in the network of cameras.

2. The inputs are the query and the tracklets of all people appearing on the network of cameras. Re-identification datasets almost always provide quasi-perfect inputs of detection and tracking.

3. This is an offline process.

As a consequence, adapting the re-identification challenge to an online framework adds several difficulties:

1. The inputs can contains errors that can completely distort an appearance-based signature, especially in the case of partial detection at the level of people detection and occlusion at the level of tracking

2. This is an online process, meaning that no assumption can be made whether a given person will reappear in the network of cameras and when.

In order to overcome these difficulties, we propose a new method called re-acquisition and re-identification method. We introduce the concept of re-acquisition as a restriction of the re-identification challenge to a single camera. The main application is to continue tracking the same person even after a long occlusion or after leaving and re-entering the scene, two cases that tracking algorithms generally cannot handled. The proposed method contains the following steps:

1. Key frames selection. This first step is used to overcome the potentially wrong inputs from the tracking results.

2. Visual signature computation. One signature is computed for each input tracklet using appearance-based feature in order to create a database of signature.

3. Similarity computation. Each newly created signature is compared with the other database signatures and a ranking of the best matching signatures is computed.

4. Tracklet matching. Using contextual information, unsupervised learning and a constrained clustering algorithm, we generate groups of tracklets representing the same person.

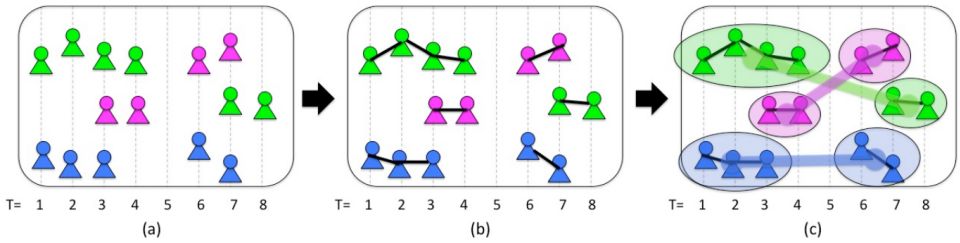


Figure 3.1: The overview of the tracking approach in T frames: (a) The raw detection results; (b) the tracking results; (c) the results of matching trajectories using online discriminative appearances

In this chapter, we use the same formalism that was introduced in 2.3.1.

3.3.2 Key frames selection

The main objective of the module is to create a reliable visual signature of each tracklet and compare them to find the ones that are the more similar. However, the reliability can only be achieved if the inputs - the tracklets, result of the tracking - are themselves reliable enough. In the case of a visual signature based on appearance, the inputs we want to avoid are the one containing missed or partial detections or occlusions. In fact, those faulty inputs will have a negative impact on all the signature computation methods described in the next subsection. That is why the first step is to extract key frames that represent the most "reliable nodes" of each tracklet. The "reliable nodes" are the ones which contain the most significant information concerning the appearance of the person with the least noise coming from interaction with the background or other people. To achieve this goal, we define an energy function for each tracklet and we try to minimize it by removing the unreliable nodes.

As stated in in the previous chapter, the feature pool $\mathcal{F}_t^i$ that characterize each node $\mathcal{C}_t^i$ of a tracklet is divided into three categories $\mathcal{F}_t^i = \{\mathcal{F}_t^{O,i}, \mathcal{F}_t^{OO,i}, \mathcal{F}_t^{OE,i}\}$:

- $\mathcal{F}_t^{O,i}$ represents the pool of features that are computed only using the data of the object (e.g. appearance, trajectory, ...)
- $\mathcal{F}_t^{OO,i}$ represents the pool of features that are computed using data of the object i considering the other objects of the scene (e.g. occlusion level, people density, ...)
- $\mathcal{F}_t^{OE,i}$ represents the pool of features that are computed using data of the object i considering the environment (e.g. occlusion level with background element, entering or leaving some zones, ...)

In order to compute the most reliable visual signature possible, we need the following features from each feature pool:

- From the Object Feature Pool $\mathcal{F}_t^{O,i}$:
  - The **appearance features** with which the visual signature will be computed. They are detailed in the next subsection.
- From the Object-Object Feature Pool $\mathcal{F}_t^{OO,i}$:
  - **Density with other objects**
  - **Spatial overlap level with other objects**
  - **Frame-to-frame overlap with other objects**

- From the Object-Environment Feature Pool $\mathcal{F}_t^{OE,i}$

  - **Object appearing/disappearing**

  - **Spatial overlap level with background elements**

The description of Object-Object Features and Object-Environment Features can be found in chapter 2.

Considering a chain of nodes $\mathcal{C}^I$ representing all the nodes with the same individual throughout all the video sequences on all cameras, the goal of the re-acquisition and re-identification method is to link the tracklets $\{\mathcal{C}^{i_1}, \mathcal{C}^{i_2}, \ldots, \mathcal{C}^{i_j}\}$ into one group.

First, for each feature f^i in the feature pool $\mathcal{F}_t^i$, we compute the mean μ:

$$\forall f^i \in \mathcal{F}_t^i = \{\mathcal{F}_t^{O,i}, \mathcal{F}_t^{OO,i}, \mathcal{F}_t^{OE,i}\} : \mu(f^i) = \frac{\sum_{t \in [T_{start}^i, T_{end}^i]} f_t^i}{|[T_{start}^i, T_{end}^i]|} \tag{3.1}$$

Then, we compute the sum of the standard deviations for each feature at each node $\sigma(f_{t_k}^i)$:

$$\forall t \in [T_{start}^i, T_{end}^i] : \sigma(f_t^i) = \frac{\sum_{f_t^i \in \mathcal{F}_t^i} |f_t^i - \mu(f^i)|}{|\mathcal{F}_t^i|} \tag{3.2}$$

Finally, the energy function $E(C^i)$ is defined as the sum of five σ:

$$E(C^i) = \{\sum_{g=1}^{5} \sigma(f_{t_g}^i)\} \tag{3.3}$$

Minimizing σ means that we select the five nodes that at the least variant regarding to their features. Considering the pool of feature used to compute σ, we obtain the five most reliable candidates for representing the tracklet and we call them key frames.

The decision to take the best five nodes is a compromise between performance and reliability 3.2. Choosing too many nodes would slow down the computation of the visual signature and choosing too few nodes could lead to lose information.

It is also to be noted that this method works under the hypothesis that a majority of the inputs are correct (the hypotheses described in section 2.3.2 are still considered) as far as the features mentioned before are concerned.

Figure 3.2: Selection of the most reliable nodes to be used as key frames

3.3.3 Signature computation

3.3.3.1 Mean Riemannian Covariance Grid descriptor

The choice of a good visual descriptor is essential for the classic re-identification challenge. As we are adapting this challenge to improve tracking results, this choice is also a key element, adding two difficulties: the descriptor has to be usable during the tracking process and resilient to errors, even if the key frame selection should have minimized this problem. The objective in this process is to create a visual signature associated to each tracklet, which is a complex data structure that represents the tracklet. This signature is paired with a similarity measure that allows us to determine how close or far two tracklets are, based on their appearance.

Mean Riemannian Covariance Grid (MRCG) descriptor [Bak *et al.* 2011] is used in the case of re-identification on overlapping or non-overlapping cameras and has already showed promising results outperforming the state of the art. This is a multi-shot approach meaning that it uses multiple images of the same person to create a signature as opposed to the single-shot approach that only make use of one image. This multi-shot approach is particularly adapted as we have access to all the tracklet images. MRCG works by forming a dense grid structure with spatially overlapping square regions (cells) 3.3. These cells are described using mean covariance matrix. Since covariance matrices do not form a vector space, this mean covariance is computed on a Riemannian manifold. The mean covariance is an intrinsic average, which blends appearances from multiple images, holding information on feature distribution, their spatial correlations and their temporal changes throughout the tracklet. As a result, each tracklet, reduced to its key frames is represented by a MRCG visual signature 3.4.

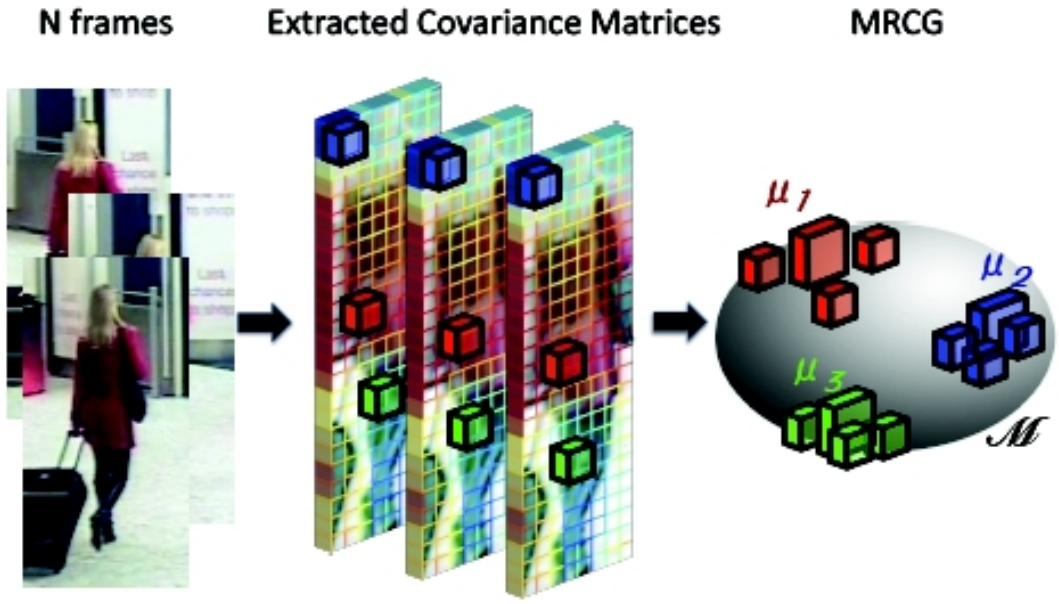


Figure 3.3: Computation of the MRCG visual signature

Figure 3.4: A MRCG signature file with the covariance coefficients for each patches of the tracklet.

3.3.4 Similarity computation

At this point, each tracklet is represented by a discriminative MRCG signature.

Using the distance associated with the MRCG signature, each time a new signature is computed, it is compared with all other signatures from the database. Inspired by the method described in [Kuo *et al.* 2010], similarities between all the combinations of signature are computed and arranged into two categories: potential matches PM and impossible matches IM. There are two conditions to decide in which category a pair of tracklet belongs. If at least one of theses conditions is met, the pair of tracklet will be considered as impossible to match:

1. Condition 1: if two tracklets have at least one node at the same time on one or multiple non overlapping cameras, the two tracklets cannot belong to the same person.
2. Condition 2: if two tracklets do not satisfy a distance constraint (e.g. minimum time required to cross an empty spot between two cameras), the two tracklets cannot belong to the same person.

In all other cases, the pair of tracklet belongs to the PM list.

The two lists PM and IM are then ranked separately from the highest similarity to the lowest 3.5.

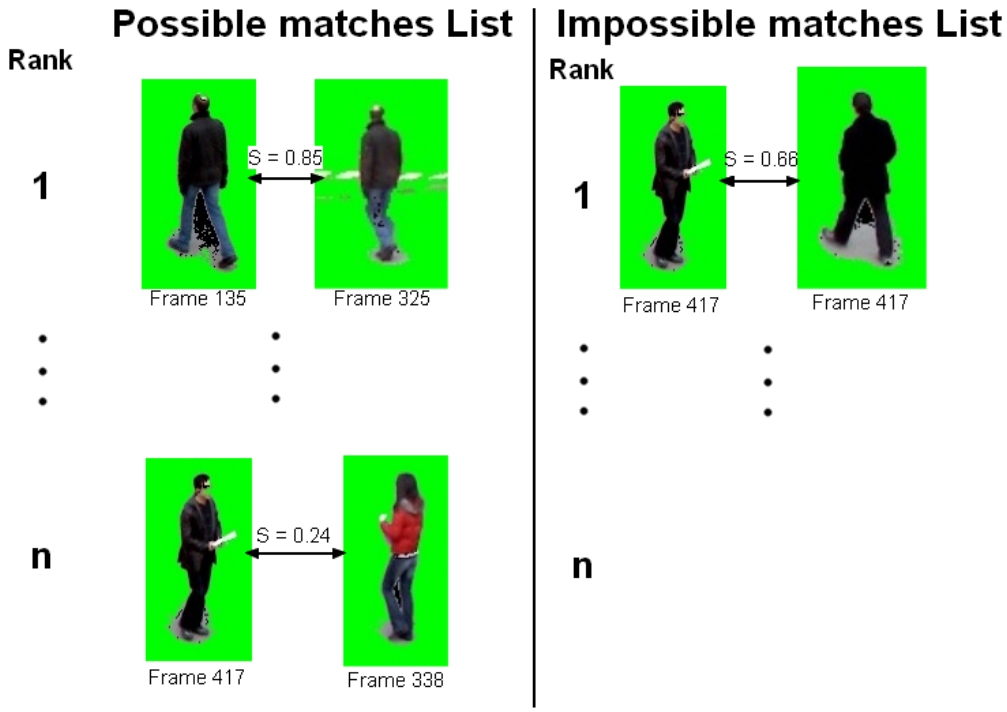


Figure 3.5: Ranking of possible matches and impossible matches

3.3.5 Tracklet matching

The list of potential matches $PM = \{pm_k\}_{k=1}^{N}$ and the list of impossible matches $IM = \{im_k\}_{k=1}^{M}$ contains all the similarities between every pairs of tracklet. Only the best potential matches are used

as candidates to be linked. To determine the best candidates, we use unsupervised learning and a constrained clustering algorithm to create the groups of tracklet corresponding to the same person.

To define these best possible matches, we consider the highest ranks of similarity between two tracklets that are impossible to link. We also introduce a parameter p called precision. It is value between 0 and the number of pair in the list of impossible matches that represents the degree of precision wanted for this method. It also represents the number of impossible matches that are used to filter the potential matches list. Using this parameter we introduce the threshold function Θ_p which is designed to keep only the most reliable candidate from the possible match list:

$$\Theta_p = \frac{1}{p} \sum_{k=1}^{p} S(im_k) \tag{3.4}$$

where S is the similarity function that compared the visual signatures and im_k is the pair at rank k of the impossible match list.

Considering this threshold, the list of potential matches is filtered to only keep the matches that have at least a similarity greater than Θ_p:

$$PM' = \{pm_k : S(pm_k) > \Theta_p\}_{k=1}^{N'} \tag{3.5}$$

For example with $p = 1$ (maximum precision), $\Theta_p = S(im_1)$, meaning that only the highest incorrect link score will be use as a threshold to decide which tracklets from the possible matches list will put together.

As defined before, the impossible match list is established online and based on a temporal constraint. The threshold Θ_p is updated with a new value if new impossible matches appear and therefore, all previous matches are updated using the new threshold. This threshold represents an adaptive parameter of the proposed approach.

The parameter p is set manually and represents the number of impossible links that are considered to compute the threshold. As the parameter p increases, the number of correct links also increases but we have also a greater chance to accept wrong matches. The threshold Θ_p is only based on the impossible match list. Its reliability depends on the number of people that are detected in the scene at the same time and their similarity. Considering that it can be inappropriate to some sequences, it is also possible to learn the threshold in an offline phase using a database of images of different people.

Starting with the first rank PL', we use a constrained clustering algorithm to create clusters of tracklets that should correspond to the same person. A constrained clustering algorithm is a clustering algorithm that adds two types of rules called must-link constraint and cannot-link constraint. In our algorithm we set the following rules:

- Must-link constraint: two tracklets representing the same object and appearing in the filtered possible match list PM' must be part of the same cluster
- Cannot-link constraint : two tracklets that are part of the impossible match list IM can never be in the same cluster

Considering an existing cluster of n tracklets $[\tau_1, \cdots, \tau_n]$, and a possible match $[\tau_a, \tau_b]$, we introduce the following matching condition:

$$\frac{1}{2n}\sum_{k=1}^{n} S([\tau_a, \tau_k]) + S([\tau_b, \tau_k]) < \Theta_p \tag{3.6}$$

If the matching condition is true, the possible match $[\tau_a, \tau_b]$ is added to the cluster. However, if the matching condition is wrong for all clusters and the tracklets τ_a or τ_b do not appear in any existing cluster, a new cluster is created with these two tracklets.

3.3.6 Results

3.3.6.1 PETS2009 dataset

We evaluate the effectiveness of the proposed approach using the public dataset PETS2009, on the particular sequence S2_L1, composed of 7 overlapping cameras recording 12 different people walking. This dataset is particularly relevant for the proposed approach because it contains a lot of occlusions, people that leave the scene and come back later and a number of people high enough to perform the linking algorithm without learning the adaptive parameter with an offline learning phase.

However, the default ground truth [Beleznai *et al.* 2011] shows 21 trajectories, using two different IDs to describe a person leaving and reentering in the scene. For performance evaluation, two different ground-truth data are used, a first one with 21 trajectories and a custom ground-truth with 12 trajectories taking into account the fact that people can reenter in the scene.

In order to compare the enhanced signatures with the original one, we evaluate the percentage of tracklets that are correctly linked, incorrectly linked and not linked according to the parameter p of the equation 3.4. A tracklet is considered as correctly linked if it is classified in a cluster representing the same person. An incorrect link occurs when the tracklet is put in a cluster representing a different person. Finally, not linked tracklets correspond to tracklets that are not assigned to any cluster whereas they should be. In our first experiement (table 3.1), we use the 21 tracklets of the ground-truth as an input of our tracklet matching method to see if we can match the people leaving and re-entering the scene.

In this ideal situation where the detections are perfect (ground-truth detections), we are able to match 87.5% of the tracklets correctly while using the key frames and only 50% without. We are unable

Method	Tracklets	p	Correctly linked	Incorrectly linked	Not linked
Visual signature with all tracklets	21	1	12.5%	0%	87.5%
	21	5	50%	0%	50%
Visual signature + key frames	21	1	62.5%	0%	37.5%
	21	5	87.5%	0%	12.5%

Table 3.1: Tracklet matching rate with and without the use of the key frame selection used on the 21 tracklets of the ground-truth data.

to achieve a 100% because of the appearance similarity between some of the actors of the scene (three of them are wearing a black coat).

In a second experiment, we use tracklets given by a short-term tracker providing short but reliable tracklets to test the tracklet matching method in a real life situation (table 3.2). It is important to note that the key frame selection reduce the number of "usable" tracklet from 129 to 76 because the other tracklets are being considered as too small to be reliable (length inferior or equal to 4 frames) or noisy.

Method	Tracklets	p	Correctly linked	Incorrectly linked	Not linked
Visual signature with all tracklets	129	1	21.7%	1.6%	76.7%
	129	5	53.5%	3.9%	42.6%
	129	10	59.7%	12.4%	27.9%
Visual signature + key frames	76	1	51.4%	0%	48.6%
	76	5	71.1%	5.2%	23.7%
	76	10	78.9%	6.5%	14.6%

Table 3.2: Tracklet matching rate with and without the use of the key frame selection used on 129 tracklets given by the short-term tracker

The results show that the tracklets with signature improved by the key frames selection are more likely to be added to a correct cluster (78.9%) compared to the normal signatures (59.7%). This step also decreases the error rate from 12.4% to 6.5%.

Some specific tracking metrics are presented in [Bernardin & Stiefelhagen 2008] and [Ellis *et al.* 2009] for PETS2009 dataset. The computation of these metrics is reported in table 3.3. However the metrics

Multiple Object Tracking Accuracy (MOTA) and *Multiple Object Tracking Precision* (MOTP) are not really adapted with the proposed method. These metrics works as follows: if the same person is described with two different ID, it is counted as one single error, not taking into account the length of both tracklets. Considering all the other possible errors (miss-detection, tracking errors), the influence of one ID switch errors does not appear clearly on these metrics. As a matter of fact, the values of these metrics are not significantly influenced enough by the proposed error recovering method. One solution to improve these metrics would have been to reconstruct the trajectory during the interval between two IDs of the same person using a trajectory optimization algorithm.

Metrics	MODA	MODP	MOTA	ATA
[Berclaz *et al.* 2009]	0.84	0.53	0.82	0.15
[Yang *et al.* 2009]	0.759	0.544	0.76	0.38
[Conte *et al.* 2010]	0.833	0.645	0.830	0.092
Short-term tracker	0.8274	0.571	0.8271	0.0998
Short-term tracker + tracklet matching	0.8287	0.573	0.8284	0.1002

Table 3.3: Some of CLEAR MOT metrics from [Ellis *et al.* 2009] for the short-term tracker with and without the tracklet matching method.

In order to evaluate successfully the performance of the proposed method, we use other evaluation metrics described in [Wu & Nevatia 2007]:

- Mostly Tracked trajectories (MT) when more than 80% of the trajectory is tracked
- Partially Tracked trajectories (PT) when between 20% and 80% of the trajectory is tracked
- Mostly Lost trajectories (ML) when less than 20% the trajectory is tracked)

Method	p	MT	PT	ML
[Chau *et al.* 2011b]	–	14.3%	57.1%	28.6%
short-term tracker	–	9.5%	57.1%	33.3%
short-term tracker + tracklet matching	1	19%	61.9%	19%
	10	23.8%	57.1%	19%

Table 3.4: Tracking performance on PETS2009 S2.L1 View_001 sequence using the original ground-truth with 21 trajectories

Method	p	MT	PT	ML
[Chau *et al.* 2011b]	–	8.3%	58.3%	33.3%
short-term tracker	–	0%	41.7%	58.3%
short-term tracker + tracklet matching	1	33.3%	41.7%	25%
	10	50%	33.3%	16.7%

Table 3.5: Tracking performance on PETS2009 S2.L1 View_001 sequence using the custom ground-truth with 12 trajectories

Table 3.4 shows the results using the original ground-truth including 21 trajectories. In this case, only occlusions or miss-detections can interrupt a tracklet. Although the short-term tracker used is not better than a State of The Art tracker based on OpenCV Kalman filter [Chau *et al.* 2011b], the tracklet matching process slightly improves the results of the short-term tracker. The results shows that some tracklets can be merged after an occlusion. The tracklets of the Kalman filter based tracker are not used as an input for the global tracker because they are not reliable enough compared to the ones provided by the short-term tracker. Table 3.5 shows the results using the custom ground-truth including 12 trajectories. In this case, people leaving and reentering the scene are considered as the same person. It is normal that trackers [Chau *et al.* 2011a] and [Chau *et al.* 2011b] have only a small percentage of MT (8.3% and 0%). However, the proposed global tracker significantly improves this percentage up to 33.3% when $p = 1$ and up to 50% when $p = 10$.

3.3.6.2 CAVIAR dataset

Caviar dataset contains 26 videos, 6 of them are used for training the detection and tracking algorithm and the remaining 20 are used for evaluation. The results are presented in table 3.6. The metrics are the same that were used in the last section : mostly tracked, partially tracked and mostly lost.

Method	MT (%)	PT (%)	ML (%)
[Xing *et al.* 2009]	84.3	12.1	3.6
[Huang *et al.* 2008]	78.3	14.7	7
[Li *et al.* 2009]	84.6	14.0	1.4
[Kuo *et al.* 2010]	84.6	14.7	**0.7**
Tracklet matching	**84.6**	9.5	5.9

Table 3.6: Tracking results on the Caviar dataset

Thanks to the tracklet matching, we are able to achieve the same percentage of mostly tracked objects (84.6%) as the State of the Art. However we have difficulties to link the tracklets that are too noisy with this method.

3.3.6.3 I-LIDS dataset

We perform two experiments on i-LIDS data with multi-cameras. The evaluation is presented in the light of matching the tracklets across disjoint camera views.

i-LIDS-AA: This dataset contains 100 individuals registered in two non-overlapping cameras. For each individual a different number of cropped images is given, forming the tracklet. Our aim is to match correctly the tracklets from the first camera with the tracklets from the second camera. Although i-LIDS-AA was originally extracted for evaluating the only person re-identification problem, this dataset can also be applied to test our approach. Assuming a regular flow of individuals from the first camera to the second camera, we successfully linked 72% of tracklets. It is worth noting that the best performance for re-identification achieved on this data reached 43% for the first rank in CMC curve [Bak *et al.* 2011].

i-LIDS-crowded: This dataset contains a dense scenario with strong occlusions and complex interactions between objects. Crowded environment makes the object detection and the object tracking very challenging. We applied our short-term tracking to obtain the tracklets from both cameras. During linking the tracklets from the first camera with the tracklets from the second camera we tuned the similarity threshold to ensure 100% precision. Finally, 33.3% of ground-truth objects were linked together across

disjoint camera views.

3.4 Conclusions

In this chapter we present a new approach for improving tracking results (a first tracking stage) using an online tracklet matching method (a second tracking stage) based on an enhanced appearance signature and a new strategy for matching tracklets. The computation of the visual appearance of the target is based on key frames, which significantly improves the quality and the reliability of tracking results, while keeping a low level of errors.

A reference set of tracklet pairs which cannot be matched is used to compute a threshold indicating a value where two appearances are significantly close to each other. Thanks to this threshold, a clustering stage regroups all similar tracklets which enable to recover tracklets which have been not associated by the first stage of tracking. This unsupervised learning stage transforms the initial ranking problem into a decision problem.

This tracklet matching approach is independent from the tracker and has been applied on several trackers for the first tracking stage. However the final tracking performance still depends on the quality of the segmentation and on the people detection algorithm as any tracking algorithm.

In an real life situation, for example video surveillance in an airport, storing numerous signatures might also be a problem in terms of storage capabilities and speed of the matching process. However, since the descriptor is based on people appearance depending on their clothes which usually change after one day, the signature database life span should not exceed one day. The processing time of the matching algorithm is between 2 and 5 frames per second for 15 tracked people in the scene. The tracklet matching process is evaluated on two benchmark datasets, PETS2009 and CAVIAR. These evaluation experiments show that this second tracking stage improves the performance of the first stage and outperforms several State of The Art trackers.

In future work, we will focus on building a more complex tracklet signature, by computing different signatures depending on the positions and postures of the person. Another improvement will consist in not averaging the signature over the whole tracklet but keeping a signature for each key frame to minimize the distance with the key frame signatures of another tracklet. This new tracklet representation will improve the performance of the matching process but may also significantly increase the processing time. Finally, online evaluation of the tracking results could also be used to automatically tune the parameters depending on estimated errors. However the main parameter is the number of key frames which is not sensitive in the already conducted experiments.

Figure 3.6: Matching algorithm inputs and output. The length of the output tracklet is the sum of the lengths of the inputs tracklets. In this example, the input tracklets length is between 8 and 120 frames and the output length is 348 frames.

4

RESULTS AND APPLICATIONS

This chapter includes results from the publication:

- *"B. Fosty, C. F. Crispim-Junior, J. Badie, F. Brémond, M. Thonnat. Event Recognition System for Older People Monitoring Using an RGB-D Camera. In ASROB 2013 - Workshop on Assistance and Service Robotics in a Human Environment, 2013."*

4.1 Datasets

In order to estimate the performance of our approach and compare our results to the state of the art, we use several public datasets. These datasets have all in common to be video surveillance datasets with fixed cameras observing people indoor or outdoor.

4.1.1 CAVIAR

CAVIAR (Context Aware Vision using Image-based Active Recognition) [1] is a dataset from 2004 containing two sets of video. The first set was filmed with a wide angle camera lens in the entrance lobby of the INRIA Labs at Grenoble, France and the second one also uses a wide angle lens along and across the hallway in a shopping center in Lisbon. All video clips are recorded with a resolution of 384x288 pixels at 25 frames per second. The ground truth is also available.

[1] CAVIAR: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

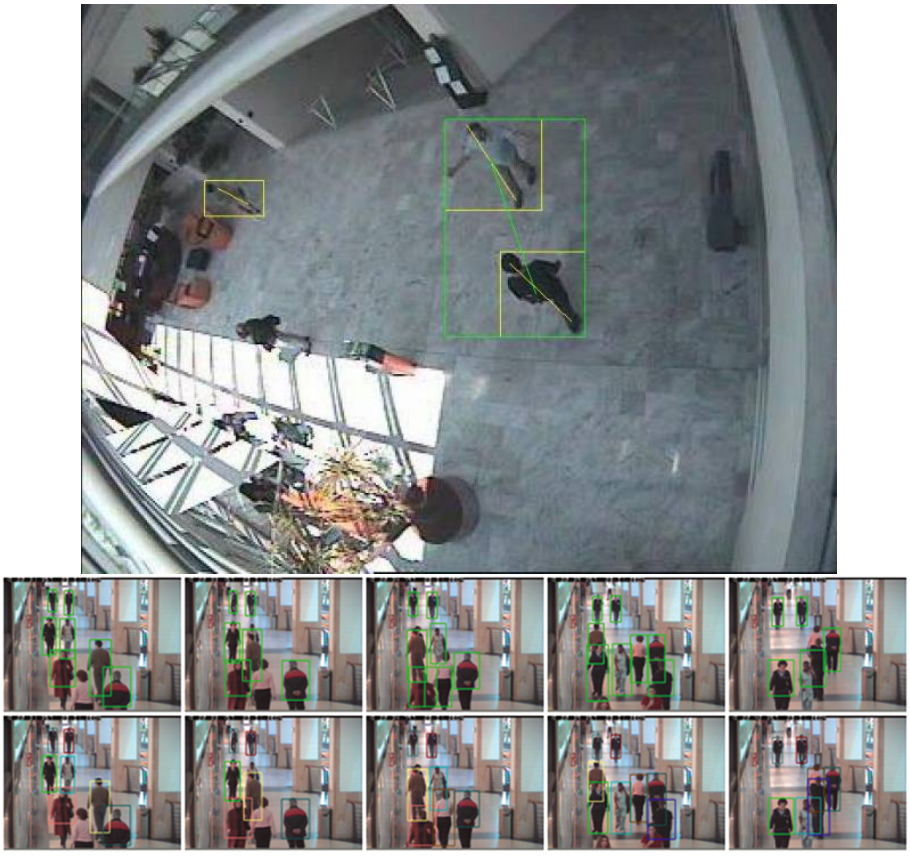


Figure 4.1: Overview of the CAVIAR dataset

4.1.2 I-LIDS

The i-LIDS library (Imagery Library for Intelligent Detection Systems) [2] is a UK government initiative to facilitate the development and evaluation of vision based detection systems which meet Government requirements. This dataset was release in 2006 and contains videos filmed at Gatwick airport during more than 20 hours on 5 different points of view. The main applications of this dataset are multiple camera tracking and event recognition.

[2] I-LIDS: https://www.gov.uk/imagery-library-for-intelligent-detection-systems

Figure 4.2: Overview of the multi-object tracking part of the I-LIDS dataset

4.1.3 PETS2009

PETS2009 (Performance Evaluation of Tracking and Surveillance) [3] is a dataset that contains multiples sequences filmed by 8 overlapping cameras at different points of view. It includes three main challenges: person count and density estimation (dataset S1), People Tracking (dataset S2) and flow analysis and event recognition (dataset S3). Each video is between between 700 and 1000 frames at 7 frames per second with a resolution of 768x576 pixels. The dataset S2 is mainly used in this thesis because it contains the main challenges we want to address: multiple people tracking (around 15 at the same time), strong occlusions and people leaving and re-entering the scene.

[3]PETS2009: http://www.cvg.reading.ac.uk/PETS2009/

Figure 4.3: Overview of the PETS2009 dataset

Figure 4.4: The 8 different views of the PETS2009 dataset

4.1.4 TUD-Stadtmitte

TUD-Stadtmitte [4] is a dataset released in 2010 which contains one sequence of people walking in pedestrian areas. The video is around 200 frames at a resolution of 640x480 pixels. This dataset is particularly challenging for people tracking and occlusion management. Two other sequences (TUD-Campus and TUD-Crossing) were added later with the same type of scenario.

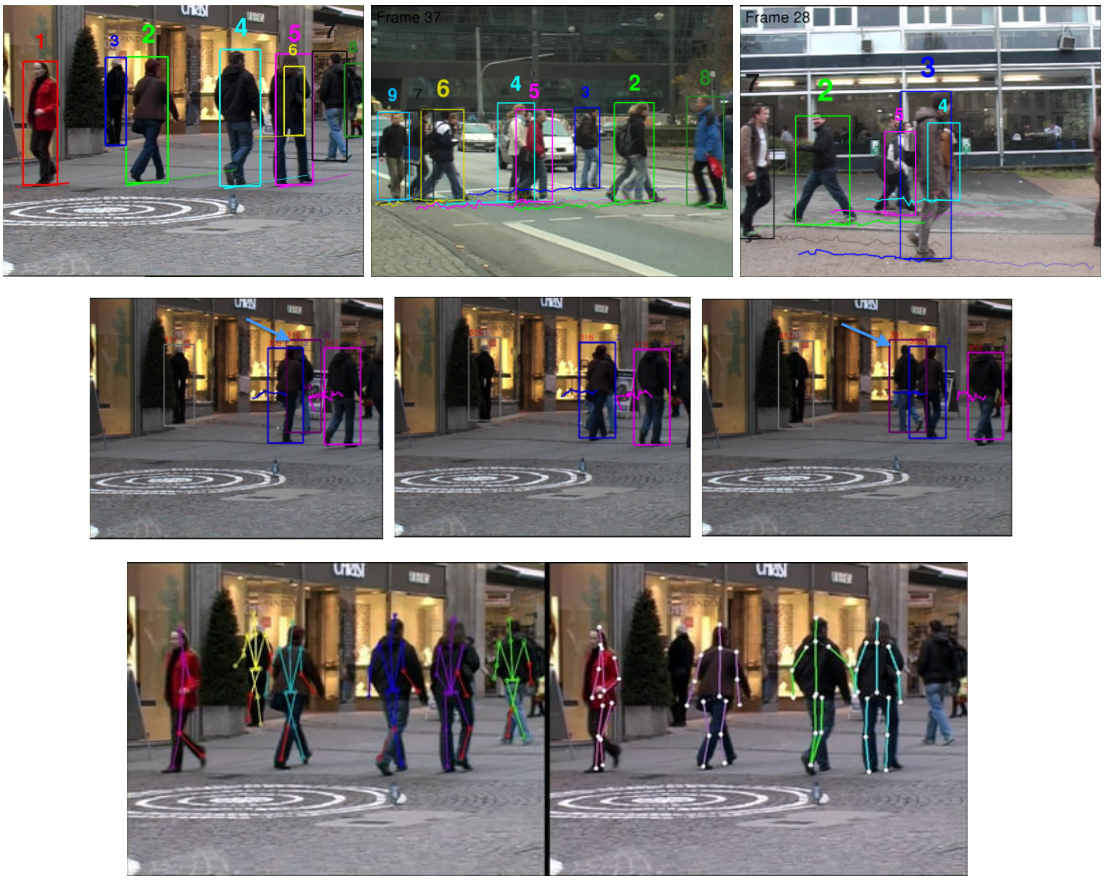


Figure 4.5: Overview of the TUD dataset

4.1.5 VANAHEIM

The VANAHEIM project [5] (Video/Audio Networked surveillance system enhAncement through HumancEntered adaptIve Monitoring) is a European project started in 2010 and finished in 2013. The aim of this project is to study innovative surveillance components for autonomous monitoring of multi-Sensory and networked infrastructure such as underground transportation environment. It contains long se-

[4]TUD-Stadtmitte: https://www.d2.mpi-inf.mpg.de/node/428
[5]VANAHEIM: http://www.vanaheim-project.eu/

quences recorded in a Torino subway station and is particularly interesting for group tracking, people counting and event detection. This dataset has not been release to the public yet.

Figure 4.6: Overview of the VANAHEIM dataset

The VANAHEIM dataset was used to experiment the reliability of the online evaluation and the tracklet matching methods when dealing with groups. The tracklet matching method was particularly efficient to retrieve lost groups after an occlusion.

4.1.6 CARETAKER project

The CARETAKER project [6] (Content Analysis and REtrieval Technologies to Apply Extraction to massive Recording) aims at studying, developing and assessing multimedia knowledge-based content analysis, knowledge extraction components and metadata management sub-systems in the context of automated situation awareness, diagnosis and decision support. The dataset contains sequences filmed in the Roma and Torino subways with more than 30 sensors each (20 cameras and 10 microphones). The main challenges of this dataset are tracking and event recognition especially abnormal behaviors (for example a person jumping over a barrier).

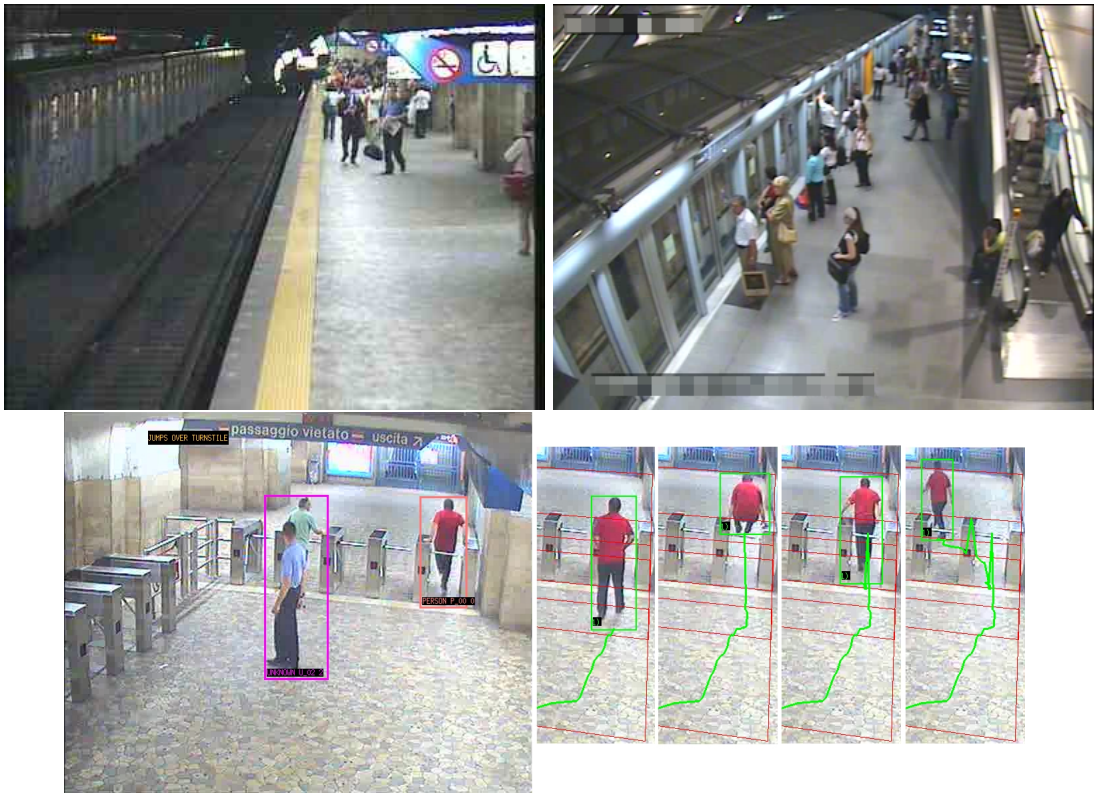


Figure 4.7: Overview of the CARETAKER dataset

4.1.7 Nice Hospital dataset

This dataset is a private dataset made by INRIA in collaboration with the Memory Center of the Nice Hospital. It contains lots of sequences of elderly people who are asked to do physical exercises in order to detect signs of Alzheimer disease. The video generally contains only one or two persons at the time.

[6] CARETAKER: http://cordis.europa.eu/ist/kct/caretaker_synopsis.html

However, the tracking needs to be near perfect in order to perform an accurate event recognition.

Figure 4.8: Overview of the Nice Hospital dataset

4.2 Implementation

All the presented algorithms are either implemented on the common platform of the Pulsar team called SUP (Scene Understanding Platform) and written in C++ or directly implemented in Python. The platform provides a modular environment where each member of the team can implement his own algorithm anywhere in the processing chain (video acquisition, segmentation, detection, tracking, event detection, etc.). As it can be difficult and time-consuming to optimize and debug a C++ code, Python is also used for prototyping some of the algorithm. We also consider that real-time is not a requirement because we are processing recorded videos so we do not necessary try to optimize the algorithm. However, none of the implemented algorithms takes more than three seconds to process one frame on a standard resolution video. The main library used in both C++ and Python is OpenCV [7]. All the results presented in this chapter and the previous chapters are obtained on a Fedora 17 64bits with 16GB RAM, even if this amount of memory is never fully required.

4.3 Global Tracker Evaluation

4.3.1 Results on the PETS2009 dataset

The results for the PETS2009 dataset are given in table 4.1. In this test, we use the CLEAR MOT metrics to compare with other tracking algorithms. The first metric is MOTA which computes Multiple Object Tracking Accuracy. The second metric is MOTP computing Multiple Object Tracking Precision. We also define a third metric $\bar{M}$ representing the average value of MOTA and MOTP. All these metrics are normalized in the interval $[0; 1]$. The higher these metrics, the better the tracking quality is.

[7] OpenCV: http://opencv.org/

Methods	MOTA	MOTP	$\bar{M}$
[Berclaz *et al.* 2011]	0.80	0.58	0.69
[Ben Shitrit *et al.* 2011]	0.81	0.58	0.70
[Henriques *et al.* 2011]	0.85	0.69	0.77
[Zamir & Shah 2012]	**0.90**	0.69	0.80
[Milan *et al.* 2013b]	**0.90**	**0.74**	**0.82**
Online evaluation	**0.90**	**0.74**	**0.82**
Tracklet matching	**0.83**	**0.68**	**0.755**
Global Tracker	**0.92**	**0.76**	**0.84**

Table 4.1: Tracking results on sequence S2.L1.View1 of the PETS2009 dataset

In this case, the online evaluation alone associated with the tracking algorithm is able to get the same performance as the State of The Art. However, the association of online evaluation and the tracklet matching outperform the State of The Art results.

4.3.2 Results on the CAVIAR dataset

The results for the CAVIAR dataset are given in table 4.2. For this dataset, we use the mostly tracked, partially tracked and mostly lost metrics.

Method	MT (%)	PT (%)	ML (%)
[Xing *et al.* 2009]	84.3	12.1	3.6
[Huang *et al.* 2008]	78.3	14.7	7
[Li *et al.* 2009]	84.6	14.0	1.4
[Kuo *et al.* 2010]	84.6	14.7	**0.7**
Online Evaluation alone	82.6	11.7	5.7
Tracklet matching	84.6	9.5	5.9
Global Tracker	**86.4**	**8.3**	5.3

Table 4.2: Tracking results on the Caviar dataset

One again, by combining the two modules of the Global Tracker, we were able to outperform the results of the State of The Art.

4.4 Online Controller

In this section, the online evaluation module is used in an online controller. The online controller has a number of preset parameters that are learned depending on the context of the scene. The online evaluation module detects if the tracking quality decreases and asks for a change of parameters.

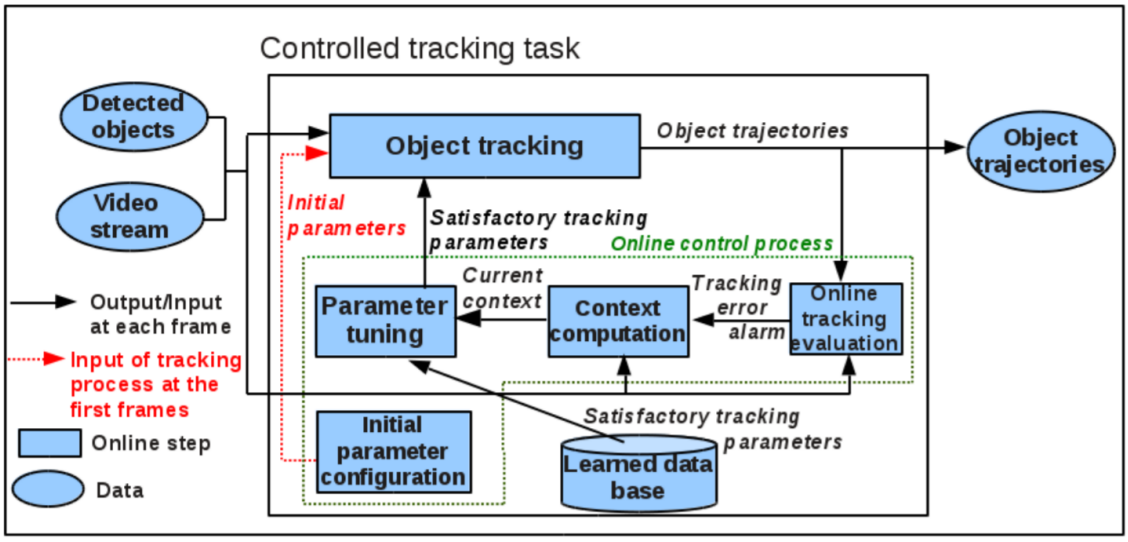


Figure 4.9: The online evaluation module inside the online controller. The online evaluation uses a codebook of tracking parameters associated to a context. This codebook is learnt offline. The online evaluation helps to compute the current context of the scene which influence the parameter tuning of the tracking algorithm.

4.4.1 Results on the Caretaker dataset

The first tested video sequence belongs to the Caretaker European project which video camera is installed in a subway station. The length of this sequence is 5 minutes. It contains 38 mobile objects.

Figure 4.10: Illustration of the output of the controlled tracking process. Different IDs represent different tracked objects.

Figure 4.10 illustrates the output of the controlled tracking process. We consider the tracking result of the two persons on the left images. At the frame 125, these two persons with respectively ID 254 (the left person) and ID 215 (the right person) are correctly tracked. Person 254 has a larger bounding box than person 215. At the frame 126, due to an incorrect detection, the left person has a quite small bounding box. By consequence, the IDs of these two persons are switched because the tracking algorithm currently uses object 2D area as an important descriptor. Now the online tracking evaluation sends an alarm on tracking error to the context computation task. At the frame 127, after the tracking parameter tuning, the two considered objects take the correct IDs as in frame 125.

Table 4.3 presents the tracking results of the tracker in two cases: without and with the proposed controller. We find that the proposed controller helps to improve significantly the tracking performance. The value of MT increases 52.7% to 84.2% and the value of ML decreases 18.4% to 10.5%.

Method	MT (%)	PT (%)	ML (%)
Tracker alone	52.7	28.9	18.4
Tracker + controller	84.2	5.3	10.5

Table 4.3: Tracking results on the Caretaker subway video. The controller improves significantly the tracking performance.

4.4.2 Results on the PETS2009 dataset

In this test, we select the sequence S2.L1, camera view 1, time 12.34 belonging to the PETS 2009 dataset for testing because this sequence is experimented in several state of the art trackers. This sequence has 794 frames, contains 21 mobile objects and several occlusion cases. For this sequence, the tracking error alarms are sent six times to the context computation task. Table 4.4 presents the metric results of the proposed approach and four recent trackers from the state of the art. With the proposed controller, the tracking result increases significantly. We also obtain the best values in all the three metrics.

Methods	MOTA	MOTP	$\bar{M}$
[Berclaz *et al.* 2011]	0.80	0.58	0.69
[Ben Shitrit *et al.* 2011]	0.81	0.58	0.70
[Henriques *et al.* 2011]	0.85	0.69	0.77
[Zamir & Shah 2012]	**0.90**	0.69	0.80
[Milan *et al.* 2013b]	**0.90**	**0.74**	**0.82**
Tracker alone	0.85	**0.74**	0.80
Tracker + controller	**0.90**	**0.74**	**0.82**

Table 4.4: Tracking results on sequence S2.L1.View1 of the PETS2009 dataset

4.4.3 Results on the TUD dataset

For the TUD dataset, we select the TUD-Stadtmitte sequence. This video contains only 179 frames and 10 objects but is very challenging due to heavy and frequent object occlusions. Table 4.5 presents the tracking results of the proposed approach and three recent trackers from the state of the art. We obtain the best MT value compared to these two trackers.

Method	MT (%)	PT (%)	ML (%)
[Kuo & Nevatia 2011]	60.0	30.0	10.0
[Andriyenko & Schindler 2011]	60.0	30.0	10.0
Tracker alone	50.0	30.0	20.0
Tracker + controller	**70.0**	10.0	20.0

Table 4.5: Tracking results on the TUD-Stadtmitte video. The controller improves significantly the tracking performance.

4.5 Application to event recognition using RGB-D camera

In many domains such as health monitoring, the semantic information provided by automatic monitoring systems has become essential. These systems should be as robust, as easy to deploy and as affordable as possible. This section presents a monitoring system for mid to long-term event recognition based on RGB-D (Red Green Blue + Depth) standard algorithms and on additional algorithms in order

to address a real world application. Using a hierarchical model-based approach, the robustness of this system is evaluated on the recognition of physical tasks (e.g. balance test) undertaken by older people during a clinical protocol devoted to dementia study. The performance of the system is demonstrated at recognizing complex events based on posture and 3D contextual information of the scene (Table 4.6).

The system architecture is presented in 4.11. The first steps of the vision component performs people detection and tracking based on the open source framework OpenNI, through NestK library. The tracking results are then improved by the Global Tracker before being processed by the Event Recognition Module. In this last module, the goal is to recognize complex events. The extraction of complex events from video sequences is performed by a combination of the RGB-D data stream, the corresponding tracking information (delivered mainly by the libraries NestK and OpenNI), the contextual objects (zones or equipment) and the event models.

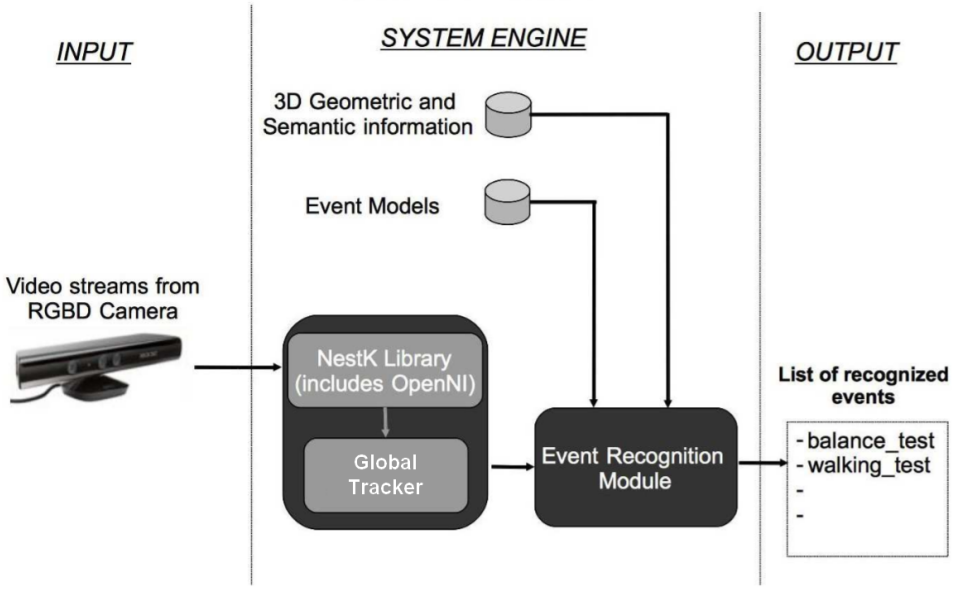


Figure 4.11: Overview of the event recognition system

The proposed system has been evaluated at monitoring the physical tasks of participants of a medical protocol for Alzheimer disease study. Participants aged more than 65 years were recruited by the Memory Center of the Nice Hospital. Participants are asked to perform a set of physical tasks and daily living activities as a basis to a clinical evaluation of their executive functions. The main scenario is intended to assess kinematic parameters about the participant's gait profile (e.g. static and dynamic balance test, walking test). During this scenario an assessor stays with the participant inside the room (see figure 4.12) and asks him/her to perform mainly four physical activities within 10 minutes (divided in sub activities). The RGB-D camera recordings are acquired at a framerate of 10 frames per seconds with an

angle of view of 57 degrees horizontally and 43 degrees vertically. These activities are briefly described as follows:

- Balance test: the participant should keep balance while performing exercises (e.g. standing with feet side by side or standing on one or the other foot).
- Walking test: the assessor asks the participant to walk through the room, following a straight path from one side of the room to another (go attempt, four meters), and then to return (return attempt, four meters).
- Repeated transfer test: The assessor asks the participant to make the first posture transfer (from sitting to standing posture) without using help of his/her arms. The examiner will then ask the participant to repeat the same action five times in a row.
- Up & go test: participants start from the sitting position and at the assessor's signal he/she needs to stand up, to walk a three meters path, to make a U-turn in the center of the room, return and sit down again.

Figure 4.12: Room where the patients evaluation take place

The following results refers to 30 videos with average time length of 6.9 minutes. Table 4.6 evaluates the system for these videos with respect to complex event recognition (5 complex events per video, 150 in total). This table shows the differences obtained for complex event recognition with and without the Global Tracker. Results obtained directly with NestK people detection output are presented on the left, while the results obtained from the proposed system are on the right.

The observed gain of performance of the proposed approach is approximately of 10% for precision, recall and F-Score. On improved version, a recall of approximately 97% is obtained on the overall activ-

ities (true positive rate) while the precision is close to 94%. This fact means that the system recognizes most of the activities from the video sequence (around 3% missed) with an acceptable amount of false positive events. For the repeated transfer test, we highlight that the improvement of the height computation of the person has improved the precision of the detection of this event, directly related to posture (from 60.4% to 90.9%). Concerning the return attempt of the Walking test, its detection is mainly improved by the use of the re-identification algorithm inside the Global Tracker.

	Only NestK		NestK + Global Tracker	
Event category	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Balance test	90.0	96.4	100	100
Walking test (go attempt)	93.3	93.3	100	90.9
Walking test (return attempt)	**73.3**	95.7	**90**	100
Repeated transfer test	96.7	**60.4**	100	**90.9**
Up & go test	80.0	85.7	93.3	90.3
Total	**86.7**	**82.8**	**96.6**	**94.2**
Global F-Score	**84.7**		**95.4**	

Table 4.6: Event recognition performance

4.6 Conclusion

The proposed Global Tracking has been evaluated on seven datasets. These results show that the addition of the Global Tracker always improve the performance of a simple tracker (tracker without post-processing). Moreover, the Global Tracker has been compared with state of the art tracking algorithms and was able to outperform their results. There are still some remaining errors mainly due to segmentation or people detection algorithms. Future work will consist in evaluating the performance of Global Tracker to new evaluation platforms : MOT Challenge [8] and Modene Challenge [Solera *et al.* 2015]. Modene Challenge is particularly interesting because it evaluates the performance of people tracker independently from the quality of people detection. The evaluation platform generates people detection outputs with various levels of error to assess how resilient is a tracker from detection errors.

[8]MOT Challenge: https://motchallenge.net/

5

CONCLUSION

5.1 Summary of contributions

This thesis addresses the challenge of improving the quality of people tracking in video by introducing a new framework called Global Tracker, designed as a post-tracking process. This framework is divided into two main parts : the evaluation of tracker performance (chapter 2) and the tracklet matching over time process (chapter 3).

Firstly, we propose a new approach for online evaluation without ground-truth (chapter 2). By using a statistical model based on several features, it is possible to detect anomalies for each detected object. These anomalies are then classified as real errors or natural phenomenon, depending on contextual information like the object neighborhood, the environment (eg. obstacles) and the scene context (eg. zones of interest). This approach is motivated by an analysis of existing methods for offline evaluation which rely on metrics and ground-truth. This analysis allows us to create a dedicated set of non-redundant features to detect most of the potential errors. This proposed framework is useful to estimate the quality of a tracker when, for example, the ground-truth is not available or when the tracker is used in a live situation.

Secondly, a re-acquisition and re-identification method is proposed (chapter 3) in order to improve the tracking quality by correcting specific errors that result in an object ID change. This method is

inspired by the State of The Art on re-identification and adapted to fit the constraints of an online system with potential errors in the input tracking data. Our approach is based on the computation and comparison of the visual signature for each tracked object, generated by thanks to an efficient visual descriptor to get the best representation of the object appearance. The best signature matches are then ranked and the decision to link two trajectories supposedly representing the same object is handled by an unsupervised learning algorithm based on a constrained clustering algorithm. The most costly task in term of processing time is the signature computation depending on the selected features. The covariance descriptor is one of the most expensive features to compute, but even with this expensive feature, the tracking process can be real-time. On high resolution images, the overhead of the Global Tracker could be at the maximum 3 to 5 frames per second.

Finally, we evaluate the Global Tracker as a whole (chapter 4). Several datasets with different kinds of scenarios and challenges are used to evaluate the Global Tracker, outperforming the results of the State of The Art trackers. A second part of this chapter focuses on showing some of the potential applications of the Global Tracker when it is embedded in a complete system. In particular, we manage to get better performance with the Global Tracker with RGB-D cameras and also in a feedback-based system.

5.2 Limitations

The Global Tracker has been evaluated on seven datasets, most of them are public ones used as benchmark for the people tracking community. The evaluation has been done in comparison with the best State of The Art algorithms at the evaluation time. The comparison has been done also with two simple trackers, the Global Tracker on top of these simple trackers and also the Global Tracker inside an online controller framework. In all these evaluations, the Global Tracker has managed to get the best performance. The main limitations are due to segmentation and people detection errors. In case of heavy occlusion, the Global Tracker may have difficulties to recover the correct track, especially when the duration of the occlusion is important, the speed of occluded people is low and the appearance of the occluded people is low contrasted.

5.3 Future work

5.3.1 Short-term Perspectives

The multiple object tracking community is very active and new tracking algorithms and new evaluation platforms (for example MOT Challenge [1]) are proposed every year. Therefore the first task consists in comparing the performance of the Global Tracker with new State of The Art people trackers such as [Dehghan *et al.* 2015]. This tracker computes a complete graph where each node is a reliable tracklet and edges are the distance between the tracklets. The performances of this tracker are impressive but the algorithm is still costly in term of processing time due to complex optimization algorithm. Concerning the new evaluation platforms, the reproducible robustness platform [Solera *et al.* 2015] is particularly interesting because it enables to asses the tracking quality independently from the people detection output.

Another interesting work to conduct is to add some features which can better characterize people appearance. In particular for RGB-D sensors, it could be interesting to add a texture feature on the depth map (eg. LBP) or to add 4D tracklet features. To cope with partial occlusions, interesting points or body parts could be very effective to still track the part of the person which is still observable. Moreover, there are many studies which have been done on mono-object tracker and which consist in discriminating the appearance of the object (defined as a template at the initialization step) from the neighboring background. Popular techniques for characterizing the object template include sparse dictionary learning and Fourier coefficients estimation (for example VOT Challenge [2]). These techniques could be applied in case of multi-object tracking [Fagot-Bouquet *et al.* 2015].

In chapter 3, the tracklet matching process computes a threshold to estimate when two people appearance are close to each other. This estimation is performed using a reference set of tracked people surrounding the target. However, people with similar appearance could be part of the reference set (situation of people with uniform) which may lead to a very small threshold, preventing to match the target with its corresponding tracklet. Therefore, an additional step is necessary to remove too similar people from the reference step and use more geometric feature to distinguish the people with similar appearance.

5.3.2 Long-term Perspectives

In chapter 2, an online evaluation process is presented which automates the detection of potential tracking errors. However, no convincing methods have been proposed to recover from these errors. Sev-

[1] MOT Challenge: https://motchallenge.net/

[2] VOT Challenge : http://www.votchallenge.net/

eral mechanisms could be envisaged. In particular, when the detection of a target starts to be incoherent with the remaining part of the tracklet, back tracking process could be launched, starting from this new detection towards the previous detections [SanMiguel *et al.* 2010]. Another recovering mechanism could consist in re-launching another tracking process with updated parameters. The challenge in this recovering mechanism is to still allow a real time process.

In many situations, several cameras could obverse one person and so many approaches have been proposed to combined the detection of the same person observed by different cameras. In chapter 3, a matching algorithm is proposed to compute the correspondences between detections of the person observed by different cameras based on appearance features. When cameras are calibrated and synchronized, this approach could take advantage of the known locations of the person and the time of the detections, to improve the matching process.

A recurrent problem in object tracking is how to tune tracking parameters depending on the processed video. The tracklet matching process relies on two parameters which are sensitive depending on the processed video. An online controller is experimented in chapter 4 which enables to characterize the context of a video and to match it with tracking parameters through a learning phase using ground-truth on reference videos. However this matching function is brittle given the large variety of video context and ways to characterize mobile objects which could differ from different tracking algorithms. Much work still needs to be done to optimize the selection and computation of the video context features depending on the most influential tracking parameters.

APPENDIX A

PUBLICATIONS

- *"J. Badie, F. Brémond. Global tracker: an online evaluation framework to improve tracking quality. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2014."*
- *"D.-P. Chau, J. Badie, F. Brémond, M. Thonnat. Online Tracking Parameter Adaptation based on Evaluation. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2013."*
- *"J. Badie, S. Bak, S.-T. Serban, F. Brémond. Recovering people tracking errors using enhanced covariance-based signatures. In PETS 2012 workshop, associated with IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2012."*
- *"S. Bak, D.-P. Chau, J. Badie, E. Corvee, F. Brémond, M. Thonnat. Multi-target Tracking by discriminative analysis on Riemannian Manifold. In IEEE International Conference on Image Processing, 2012."*
- *"B. Fosty, C. F. Crispim-Junior, J. Badie, F. Brémond, M. Thonnat. Event Recognition System for Older People Monitoring Using an RGB-D Camera. In ASROB 2013 - Workshop on Assistance and Service Robotics in a Human Environment, 2013."*
- *"C. Garate, S. Zaidenberg, J. Badie, F. Brémond. Group Tracking and Behavior Recognition in Long Video Surveillance Sequences. In Computer Vision, Imaging and Computer Graphics Theory and Applications, 2014."*

BIBLIOGRAPHY

[Andriyenko & Schindler 2011] A. Andriyenko and K. Schindler. *Multi-target Tracking by Continuous Energy Minimization*. In IEEE International Conference on Advanced Video and Signal based Surveillance, 2011. 76

[Bak *et al.* 2011] S. Bak, E. Corvée, F. Brémond and M. Thonnat. *Multiple-shot Human Re-Identification by Mean Riemannian Covariance Grid*. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, pages 179–184, Klagenfurt (Austria), August 2011. 37, 52, 60

[Beleznai *et al.* 2011] C. Beleznai, D. Schreiber and M. Rauter. *Pedestrian detection using GPU-accelerated multiple cue computation*. In IEEE International Conference on Computer Vision and Pattern Recognition Workshops, pages 58–65, June 2011. 56

[Ben Shitrit *et al.* 2011] H. Ben Shitrit, J. Berclaz, F. Fleuret and P. Fua. *Tracking multiple people under global appearance constraints*. In IEEE International Conference on Computer Vision, 2011. 43, 72, 76

[Berclaz *et al.* 2009] J. Berclaz, F. Fleuret and P. Fua. *Multiple object tracking using flow linear programming*. In Performance Evaluation of Tracking and Surveillance (PETS-Winter), pages 1–8, Dec 2009. 58

[Berclaz *et al.* 2011] J. Berclaz, F. Fleuret, E. Turetken and P. Fua. *Multiple Object Tracking Using K-Shortest Paths Optimization*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011. 43, 72, 76

[Bernardin & Stiefelhagen 2008] K. Bernardin and R. Stiefelhagen. *Evaluating multiple object tracking performance : the CLEAR MOT metrics*. In EURASIP Journal on Image and Video Processing, volume 2008, pages 1:1–1:10, Jan 2008. 27, 57

[Bernardin *et al.* 2006] K Bernardin, A. Elbs and R. Stiefelhagen. *Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment*. In IEEE International Workshop on Visual Surveillance, May 2006. 31

[Boom *et al.* 2012] B.J. Boom, P.X. Huang, Jiyin He and R.B. Fisher. *Supporting ground-truth annotation of image datasets using clustering*. In International Conference on Pattern Recognition, Nov 2012. 25

[Boom *et al.* 2013] B.J. Boom, P.X. Huang and R.B. Fisher. *Approximate Nearest Neighbor Search to Support Manual Image Annotation of Large Domain-specific Datasets*. In International Workshop on Video and Image Ground Truth in Computer Vision Applications, 2013. 25

[Chau *et al.* 2009] D Chau, F. Brémond, E Corvée and M. Thonnat. *Repairing People Trajectories based on Point Clustering*. In International Conference on Computer Vision Theory and Applications, pages 449–455, 2009. 48

[Chau *et al.* 2011a] D. P. Chau, F. Brémond and M. Thonnat. *A multi-feature tracking algorithm enabling adaptation to context variations*. In International Conference on Imaging for Crime Detection and Prevention, Nov 2011. 59

[Chau *et al.* 2011b] D. P. Chau, F. Brémond, M. Thonnat and E. Corvée. *Robust Mobile Object Tracking Based on Multiple Feature Similarity and Trajectory Filtering*. In International Conference on Computer Vision Theory and Applications, 2011. 59

[Conte *et al.* 2010] D. Conte, P. Foggia, G. Percannella and M. Vento. *Performance Evaluation of a People Tracking System on PETS2009 Database*. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, pages 119–126, September 2010. 58

[Dehghan *et al.* 2015] A. Dehghan, S. M. Assari and M. Shah. *GMMCP-Tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking*. In IEEE International Conference on Computer Vision and Pattern Recognition, 2015. 83

[Ellis *et al.* 2009] A. Ellis, A. Shahrokni and J.M. Ferryman. *PETS2009 and Winter-PETS 2009 results: A combined evaluation*. In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), pages 1 –8, December 2009. 15, 57, 58

[Fagot-Bouquet *et al.* 2015] L. Fagot-Bouquet, R. Audigier, Y. Dhome and F. Lerasle. *Collaboration and spatialization for an efficient multi-person tracking via sparse representations*. In IEEE International Conference on Advanced Video and Signal-based Surveillance, 2015. 83

[Henriques *et al.* 2011] J. F. Henriques, R. Caseiro and J. Batista. *Globally optimal solution to multi-object tracking with merged measurements*. In IEEE International Conference on Computer Vision, 2011. 43, 72, 76

[Huang *et al.* 2008] C. Huang, B. Wu and R. Nevatia. *Robust object tracking by hierarchical association of detection responses*. In IEEE European Conference on Computer Vision, 2008. 44, 60, 72

[Kasturi *et al.* 2009] R Kasturi, D. Goldgof and P. et al. Soundararajan. *Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol Pattern Analysis and Machine Intelligence*. In IEEE Transactions on In Pattern Analysis and Machine Intelligence, 2009. 27, 30

[Kuo & Nevatia 2011] C.-H. Kuo and R. Nevatia. *How does person identity recognition help multi-person tracking?* In IEEE International Conference on Advanced Video and Signal based Surveillance, 2011. 76

[Kuo *et al.* 2010] C.-H. Kuo, C. Huang and R. Nevatia. *Multi-target tracking by on-line learned discriminative appearance models*. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 685–692, June 2010. 44, 48, 54, 60, 72

[Li *et al.* 2009] Y. Li, C. Huang and R. Nevatia. *Learning to associate: Hybridboosted multi-target tracker for crowded scene*. In IEEE International Conference on Computer Vision and Pattern Recognition, 2009. 44, 60, 72

[Maggio & Cavallaro 2010] E. Maggio and A. Cavallaro. Video tracking: Theory and practice. Wiley, 2010. 23

[Milan *et al.* 2013a] A. Milan, K. Schindler and S. Roth. *Challenges of Ground Truth Evaluation of Multi-Target Tracking*. In IEEE International Conference on Computer Vision and Pattern Recognition - Workshops, 2013. 25, 30

[Milan *et al.* 2013b] A. Milan, K. Schindler and S. Roth. *Detection and Trajectory-Level Exclusion in Multiple Object Tracking*. In IEEE International Conference on Computer Vision and Pattern Recognition, 2013. 43, 72, 76

[Nghiem *et al.* 2007] A. T. Nghiem, F. BrÃ©mond, M. Thonnat and V Valentin. *ETISEO, performance evaluation for video surveillance systems*. In IEEE International Conference on Advanced Video and Signal-Based Surveillance, pages 476–481, 2007. 27

[SanMiguel *et al.* 2010] J.C. SanMiguel, A. Cavallaro and J.M Martinez. *Adaptive on-line performance evaluation of video trackers*. In IEEE Transactions on Image Processing, 2010. 84

[Sinha 2006] P Sinha. *Face recognition by humans: Nineteen results all computer vision researchers should know about*. In Proceedings of the IEEE, pages 1948–1962, 2006. 47

[Smith *et al.* 2005] K. Smith, D. Gatica-Perez, J. Odobez and Sileye Ba. *Evaluating Multi-Object Tracking*. In IEEE Conference on Computer Vision and Pattern Recognition - Workshops, June 2005. 31

[Solera *et al.* 2015] F. Solera, S. Calderara and R. Cucchiara. *Towards the evaluation of reproducible robustness in tracking-by-detection*. In IEEE International Conference on Advanced Video and Signal-based Surveillance, 2015. 79, 83

[Szczodrak *et al.* 2010] M. Szczodrak, P. Dalka and A. Czyzewski. *Performance evaluation of video object tracking algorithm in autonomous surveillance system*. In International Conference on Information Technology, pages 31–34, June 2010. 31

[Vondrick *et al.* 2013] C. Vondrick, D. Patterson and D. Ramanan. *Efficiently Scaling up Crowdsourced Video Annotation*. International Journal of Computer Vision, pages 1–21, 2013. 25

[Wu & Nevatia 2006] B Wu and R Nevatia. *Tracking of Multiple, Partially Occluded Humans based on Static Body Part Detection*. In IEEE International Conference on Computer Vision and Pattern Recognition, 2006. 27

[Wu & Nevatia 2007] B. Wu and R. Nevatia. *Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors*. International Journal of Computer Vision, vol. 75, no. 2, pages 247–266, Nov 2007. 48, 58

[Xing *et al.* 2009] J. Xing, H. Ai and S. Lao. *Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses*. In IEEE International Conference on Advanced Video and Signal-based Surveillance, 2009. 44, 60, 72

[Yang *et al.* 2009] J. Yang, Z. Shi, P. Vela and J. Teizer. *Probabilistic multiple people tracking through complex situations*. In IEEE Conference on Computer Vision and Pattern Recognition - Workshops, pages 1 –8, Dec 2009. 58

[Yin *et al.* 2007] F. Yin, D. Makris and S. A. Velastin. *Performance Evaluation of Object Tracking Algorithms*. In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Oct 2007. 31

[Zamir & Shah 2012] A. Zamir A. R. Dehghan and M. Shah. *GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs*. In IEEE European Conference on Computer Vision, 2012. 43, 72, 76