View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

Global tracker: an online evaluation framework to improve tracking quality

Julien Badie, François Bremond

▶ To cite this version:

Julien Badie, François Bremond. Global tracker: an online evaluation framework to improve tracking quality. AVSS - 11th IEEE International Conference on Advanced Video and Signal-Based Surveillance, Aug 2014, Seoul, South Korea. hal-01062766

HAL Id: hal-01062766

https://hal.inria.fr/hal-01062766

Submitted on 10 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Global tracker: an online evaluation framework to improve tracking quality

Julien Badie and François Brémond
INRIA Sophia Antipolis, STARS group
2004 route des Lucioles - BP93 06902 Sophia Antipolis Cedex - France
{julien.badie | francois.bremond} @inria.fr

Abstract

Evaluating the quality of tracking outputs is an important task in video analysis. This paper presents a new framework for estimating both detection and tracking quality during runtime. If anomalies are detected in the tracking output results, they are categorized as natural phenomena or real errors using contextual information. As this framework should be generic and work on any kind of system (single camera, camera network), a re-acquisition step using a constrained clustering algorithm is also performed in order to keep track of the object even if it leaves the scene and comes back or appears on another camera. The framework is evaluated on two datasets using different kinds of tracking algorithms.

1. Introduction

People tracking has become a very important task with various applications such as video surveillance, people monitoring or video games. However, even in the same field of application, there is an infinite number of possible scenarios, environments and contexts. Thus, no detection nor tracking algorithm can perform perfectly in all situations. Performance evaluation is crucial in estimating the quality of an algorithm on a given scenario or application. Maggio and Cavallaro [11] categorize two kinds of methods for performance evaluation: analytical methods and empirical methods. 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 outputs 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. In this paper, we propose a new evaluation framework based on an empirical standalone methods without using ground truth.
Choosing an evaluation framework that does not require ground truth has many different applications. One of them is giving feedback to the tracking algorithm that can tunes its own parameters to improve the results on the next frame. Another convenient application is to filter the reliable information from the tracking algorithm that can be used for a next processing step such as event detection or re-identification. In this paper, re-identification is used as an example of application of the online evaluation framework.

2. Related work

Much work has already been done on online evaluation frameworks. Some of these approaches are based on estimating the quality of the detected moving regions. For example, Erdem et *al.* [5] have proposed a method for segmentation evaluation based on contrast at the boundary of each object. Other approaches are based on trajectory ([6], [14]), time-reversibility [15], uncertainty ([12]) or different sets of features ([8]). However most of these approaches are very dependent on the given detection or tracking algorithms and are generally dedicated either to detection or tracking errors.
In this paper, we try to overcome these limitations by proposing an evaluation framework that is independent from both detection and tracking algorithms that will detect and classify anomalies. Anomalies are defined as incoherences found in the tracking output. This new framework is called global tracker because it acts as an extension of the tracking algorithm while using a larger data pool and providing a more global approach of the scene understanding. It acts as a bridge between the tracking algorithm and the next processing step. An important point of the global tracker is the genericity, as it would work with every tracking algorithms and in any kind of situation (one camera, network of overlapping or non-overlapping cameras, 3D cameras, ...). Concerning the method used, it is similar to the one used in [1] where the tracking algorithm performs data matching of different frames to create an optimized graph. However this approach is limited by the fact it seems to work only with good detection input and only uses data in a time window of 20 frames. We propose to evaluate the global tracker framework by comparing the anomalies

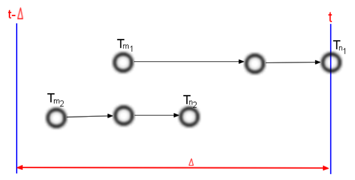


Figure 1. The tracklet representation within the time window of the global tracker

found by the online evaluation framework with the errors detected by a traditional method using the ground truth of several datasets and by showing the improvements of the results using the proposed approach.
The rest of the paper is organized as follows: Section 3 gives an outline of the framework and Section 4 explains the evaluation methodology of the global tracker and shows some results.

3. Outline of the proposed approach

The global tracker is an online process. It is defined on a sliding window of size Δ and has 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 global tracker is to update the previous knowledge of the framework $\mathcal{K}_{t-1}$ using the output data of the tracking algorithm $\mathcal{O}_t$.

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

3.1. Tracklets

The output data of the tracking algorithm $\mathcal{O}$ is 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 $[T_{m_i}, T_{n_i}]$ (fig. 1). Each node $\mathcal{C}_t^i$ corresponds to one detected object at time t and contains a pool of features $\mathcal{F}_t^i$ (eg. localization, appearance, ...). An oriented chain can have multiple parallel sub-chains, one per camera.

The main goal of the global tracker is to assess the quality of each tracklet by studying its behavior and its evolution throughout time. The global tracker is divided into two main tasks, each working at a different level and having a different goal:

- the *Interpolation* module works at the tracklet level and aims at correcting small tracking errors when one or more consecutive detections are missing in a graph, meaning that some nodes are missing in the tracklet.

- the *Tracking quality estimation* module works at the list of tracklets level and aims at detecting any anomaly found in the output of the tracking algorithm

- the *Re-acquisition and re-identification* module works on the knowledge level and aims at combining tracklets separated in time or that appear in different cameras and represent the same object. In this paper, this module is added to show the possible applications of the online evaluation.

3.2. Interpolation

Some frames may be missing in an object trajectory. It happens if the tracking algorithm fails to find a correct matching 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.

$$\exists t \in [T_{m_i}, T_{n_i}] : \mathcal{F}_t^i = \emptyset \Rightarrow \mathcal{F}_t^i = \frac{\mathcal{F}_{t-1}^i + \mathcal{F}_{t+1}^i}{2} \tag{2}$$

where $\mathcal{F}_{t-1}^i \neq \emptyset$ and $\mathcal{F}_{t+1}^i \neq \emptyset$
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. This optimization step has two main goals. The first goal is to slightly improve the tracking results because some metrics (for example the CLEAR metrics) are very sensitive to the number of missing frames in a tracking output. On the other side, it will be easier for the second module to estimate the overall quality of each tracklet if no node is missing.

3.3. Tracking quality estimation and anomaly correction

The goal of the second module is to compute control features and analyze their variations. Depending on the results, some anomalies are found. These anomalies can be real errors (ID switch, merging or splitting of two tracklets), or natural phenomena (person leaving the scene, occlusion with background elements). This module aims at determining the impact of the anomalies.

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, ...)

In order to monitor the behavior of one tracklet, the feature pool $\mathcal{F}^{O,i}$ is the most relevant. The other feature pool $\mathcal{F}^{OO,i}$ and $\mathcal{F}^{OE,i}$ are used to define the neighborhood of the object and to classify the anomalies found. For each feature $f^i \in \mathcal{F}^{O,i}$, we compute the weighted mean $\mu(f^i)$ and the weighted standard deviation $\sigma(f^i)$.

$$\mu(f^i) = \frac{\sum_{t=T_{m_i}}^{T_{n_i}} w(t) * f_t^i}{\sum_{t=T_{m_i}}^{T_{n_i}} w(t)} \tag{3}$$

$$\sigma(f^i) = \sqrt{\frac{\sum_{t=T_{m_i}}^{T_{n_i}} w(t) * (f_t^i - \mu(f^i))^2}{\sum_{t=T_{m_i}}^{T_{n_i}} w(t)}} \tag{4}$$

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

$$c(f^i) = \frac{\sigma(f^i)}{\mu(f^i)} \tag{5}$$

The potential anomalies are then detected by comparing the coefficients of variation at frame t and frame $t-1$.

$$\delta^i = \left| 1 - \frac{c(f^i)_t}{c(f^i)_{t-1}} \right| \tag{6}$$

If δ^i is near or equal to zero, it means that the last node of the tracklet keeps the same behavior as the other 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 was set to know if the value of δ could indicate 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 features used, contextual information such as entering or leaving zones in the scene can easily discriminate normal phenomena from real errors. Section 4 defines all the pools of features used for the experiment and their influence on the anomaly categorization.

If the anomaly is detected as a normal phenomenon, the node is definitively added to the tracklet. On the other hand, if an error is proven, the last node is removed from the tracklet and set in a newly created tracklet. However the risk of this algorithm is to create a set of small yet reliable tracklets. To rectify this situation, a re-identification algorithm can be used as an application of the online evaluation.

3.4. Re-acquisition and re-identification

The last module is called re-acquisition and re-identification module. The goal of re-acquisition is to merge tracklets that represent the same object but were considered as different because of long term occlusion or because the object left and re-entered the scene (fig. 2). Re-identification is a well-known problem in video surveillance where the goal is to detect one object moving in a camera network. Re-acquisition and re-identification represent the same challenge and are addressed the same way by merging several tracklets together. Merging tracklets means that their IDs i become the same and that their chain of nodes and their feature pools are also merged.

The method to merge tracklets is a constrained clustering algorithm using reliable visual features and the distance associated to these features to create clusters. The constrained clustering algorithm works as a normal clustering algorithm except that two types of constraints are added: the must-link constraints and the cannot-link constraints. These constraints create exceptions in the clustering algorithm and guarantee the integrity of the algorithm. In this module the following constraints are used:

- must-link constraints: two tracklets that were merged at time $t-1$ stay merged at time t. This constraint is based on the assumption that the online evaluation is able to provide reliable tracklets that can be easily merged together by this algorithm.
- cannot-link constraints: spatio-temporal constraints

are considered based on the camera network and the known context. For example on a single camera, two tracklets appearing on the same frame cannot represent the same object. On a non-overlapping camera network, the same object cannot be detected on two cameras at the same time.

The global tracker uses a mean-shift clustering algorithm with the above constraints and merges the tracklets according to the distance between their visual covariance descriptor described in the next section.

4. Experiments

4.1. Features used

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. It is our choice to use the same set of features for all the datasets to keep the global tracker 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 paper. Table 1 sums up the features used for the experiments.

Object features

Bounding box dimensions: The width and height of the tracked object are computed. This feature is sensitive to huge 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 [2]: 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.

Object versus object features

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:

$$density = \frac{Area(C_t^1) + Area(C_t^2)}{Area(C_t^1 \cup C_t^1)} \quad (7)$$

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:

$$spatialOverlap = max\left(\frac{C_t^1 \cap C_t^2}{Area(C_t^1)}, \frac{C_t^1 \cap C_t^2}{Area(C_t^2)}\right) \quad (8)$$

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:

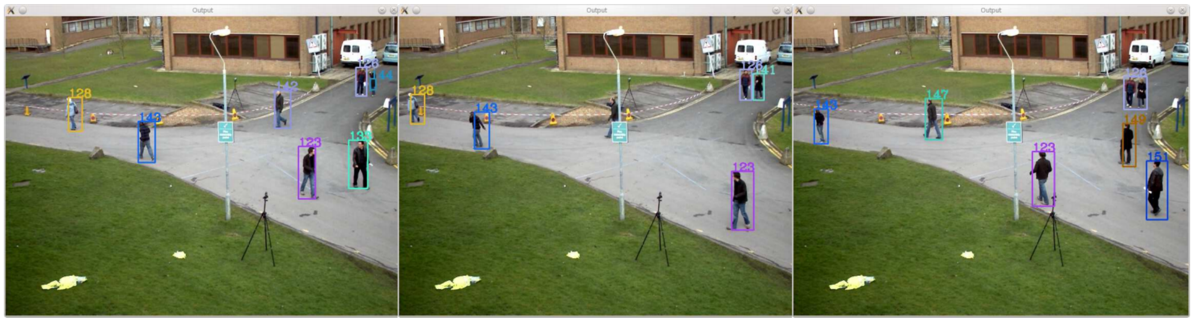
$$f2fOverlap = \frac{C_t^1 \cap C_{t-1}^2}{Area(C_t^1)} \quad (9)$$

Object versus environment features

The following features are part of the object versus environment feature pool $\mathcal{F}_t^{OE,i}$. They can be computed depending on the dataset meta-data. Offline, 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. 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 error 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.



(a) frame 194 (b) frame 213 (c) frame 233

Figure 2. The re-acquisition challenge: correcting errors due to occlusions (ID 142 on the first frame becomes 147 on the last frame) and track people that are leaving the scene and re-entering (ID 133 on the first frame becomes 151 on the last frame

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 1. Features used for experimentation

Methods	MOTA	MOTP
Berclaz et al. [4]	0.80	0.58
Shitrit et al. [3]	0.81	0.58
Henriques et al. [7]	0.85	0.69
Zamir et al. [1]	**0.90**	0.69
Milan et al. [13]	**0.90**	**0.74**
Tracker 1	0.62	0.63
Tracker 1 + global tracker	0.85	0.71
Tracker 2	0.85	**0.74**
Tracker 2 + global tracker	**0.90**	**0.74**

Table 2. Tracking results on sequence S2.L1.View1 of the PETS2009 dataset

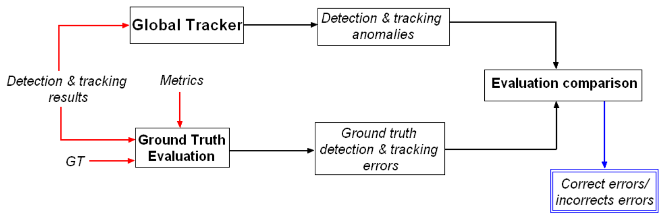

Global Tracker
Detection & tracking anomalies
Detection & tracking results
Metrics
Evaluation comparison
Ground Truth Evaluation
Ground truth detection & tracking errors
GT
Correct errors/ incorrects errors

Figure 3. The methodology to evaluate the proposed approach is based on the comparison with the ground truth

4.2. Results

The general methodology to evaluate the proposed approach is described in figure 3.

The online evaluation framework is evaluated on the public datasets PETS2009 and Caviar. 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**).

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 gives the results of the tracking algorithm with and without the global tracker with the two trackers.

The results show that while tracker 1 has difficulty providing reliable output results, the global tracker 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 global tracker 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 global tracker 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

Method	MT (%)	PT (%)	ML (%)
Li et al. [10]	84.6	14.0	1.4
Kuo et al. [9]	84.6	14.7	**0.7**
Tracker 1	78.3	16.0	5.7
Tracker 1 + global tracker	**86.4**	**8.3**	5.3

Table 3. Tracking results on the Caviar dataset

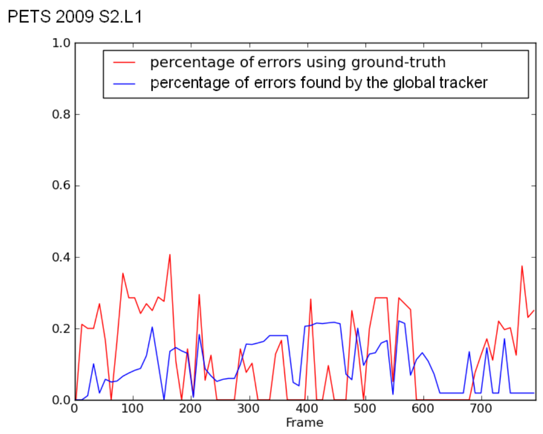


Figure 4. Comparison between the percentage of ground truth errors per frame (red) and the percentage of errors found by the global tracker (blue) on sequence S2.L1.View1 of PETS2009 using tracker 1

that less than 20% of the trajectory is correctly tracked. Table 3 gives the results of the tracking algorithm with and without global tracker with trackers 1.

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

The figure 4 shows the comparison between the percentage of errors found by the global tracker and the percentage of ground truth errors found in the tracking output for sequence S2.L1.View1 of PETS2009. For the most part, the global tracker is able to successfully detect and classify the errors (both curbs match). However it fails at the beginning of the video (around frame 75) because the global tracker has difficulties to detect errors from a tracklet that has the same error since its appearance (for example, when a background element becomes tracked).

5. Conclusions and future work

In this paper, we have presented a new framework for evaluating the quality of people trajectories during runtime without ground truth. This framework called global tracker is able to determine if anomalies appear in the tracking output. It can then classify these anomalies into errors or normal phenomena. These results are applied to a constrained clustering algorithm to handle long-term occlusions and people leaving and re-entering the scene. The global tracker is tested on two datasets and shows promising results.
The next step of this global tracker is to select and choose the more adequate set of features automatically. A feedback to the tracking algorithm could also be provided in the case of a self-tuning algorithm. A future version of the global tracker will also be able to determine during runtime which tracking algorithm is the best for the current sequence.

Acknowledgments

This work has been supported by the Conseil Regional of Provence-Alpes-Côte d'Azur (PACA), France.

References

[1] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Multiple-shot Human Re-Identification by Mean Riemannian Covariance Grid. In *AVSS*, 2011.

[2] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011.

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 2011.

[4] C. Erdem, B. Sankur, and A. Tekalp. Performance measures for video object segmentation and tracking. *IEEE Transactions on Image Processing*, 2004.

[5] D. Hall. Automatic parameter regulation of perceptual systems. *Image and Vision Computing*, 2006.

[6] J. F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV*, 2011.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *ICPR*, 2010.

[8] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by on-line learned discriminative appearance models. In *CVPR*, 2010.

[9] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009.

[10] E. Maggio and A. Cavallaro. *Video Tracking: Theory and Practice*. Wiley, 2010.

[11] E. Maggio, M. Taj, and A. Cavallaro. Efficient multitarget visual tracking using random finite sets. *CSVT*, 2008.

[12] A. Milan, K. Schindler, and S. Roth. Detection and trajectory-level exclusion in multiple object tracking. In *CVPR*, 2013.

[13] C. Piciarelli, G. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *AVSS*, 2005.

[14] H. Wu, A. C. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *PAMI*, 2010.

[15] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012.