Evaluation of Tracking Algorithm Performance without Ground-Truth Data

C. Spampinato, S. Palazzo, D. Giordano
Department of Electrical, Electronics and Computer Engineering
University of Catania, Viale Andrea Doria, 6 - 95127 Catania, Italy
Email: {cspampin,palazzosim,dgiordan}@dieei.unict.it

***Abstract*—Visual tracking is a topic on which a lot of scientific work has been carried out in the last years. An important aspect of tracking algorithms is the performance evaluation, which has been carried out typically through hand-labeled ground-truth data. Since the manual generation of ground truth is a time-consuming, error-prone and tedious task, recently many researchers have focused their attention on self-evaluation techniques for performance analysis. In this paper we propose a novel tool that enables image processing researchers to test the performance of tracking algorithms without resorting to hand-labeled ground truth data. The proposed approach consists of computing a set of features describing shape, appearance and motion of the tracked objects and combining them through a naive Bayesian classifier, in order to obtain a probability score representing the overall evaluation of each tracking decision. The method was tested on three different targets (vehicles, humans and fish) with three different tracking algorithms and the results show how this approach is able to reflect the quality of the performed tracking.**

I. Introduction

In the automatic video analysis, key roles are played by object detection, tracking and recognition (e.g. [1], [2], [3]). Object tracking consists of following an object in the video across consecutive frames; in other words, a tracking algorithm has to be able to recognize that two regions in two different frames represent the same object. Many different approaches have been studied on how to solve object tracking, from the widely-used algorithms based on Kalman filters [4] or particle filters [5] to the most recent ones based on multiple learning instance [6] or Level Sets [7]. Although the newest tracking approaches are very reliable and powerful, there is a major problem when dealing with tracking evaluation, i.e. the creation of ground truth necessary to train, test, and compare the performance of tracking algorithms. Indeed ground-truth generation is very time-consuming, error-prone and tedious to users, who basically have to analyse manually each frame of a video and label each association between objects. For this reason, some research groups are putting efforts on developing self-evaluation-based approaches, which typically evaluate tracking decisions by analyzing how regularly and smoothly an object moves (for example, a sudden change of direction of an object is considered an indication of bad tracking) or how its appearance changes (e.g. big variations in the shape ratio or in the histogram may indicate that the algorithm lost the object and is following a wrong one). The existing approaches can be classified into three main categories: 1) *Feature-based* [8] that analyse the internal state or output (shape ratio, area, speed, color and direction variations) of tracking algorithms, 2) *Hybrid-based* [9], [10] that combine several temporal and non-temporal features to get an assessment of each tracking decision and 3) *Trajectory-based* [11] that exploit intrinsic information of the generated trajectories to measure the quality of a track. The existing approaches show two main limitations: *domain-dependence*, since most of the existing approaches [8], [9], [10] identify empirically the features to be used and their contribution to the final evaluation, and *algorithm-dependence*: some approaches [11] use a-priori knowledge on the algorithm to be tested, making the method dependent on the specific application.

In this paper we propose an on-line method to test tracking algorithms without ground-truth data that analyses the regularity of motion, shape and appearance of each tracking decision and combines this information through a naive Bayesian classifier, in order to obtain a probability score representing the overall evaluation of that tracking decision. The results show how this approach is able to reflect the performance of tracking algorithms on different target motion patterns: vehicles that show a highly constrained 2D motion; people, who have more degrees of freedom than vehicles, but tend to move in 2D in a regular way, and fish, which have a typical erratic movement in 3D which is not constrained in any directions. Moreover, the use of the Bayes classifier has allowed us to establish the contribution of each feature to the final evaluation score, indicating that motion-based features do not allow to distinguish good tracking decisions from bad ones.

The remainder of the paper is as follows: in Section II we present our approach for online empirical evaluation of tracking algorithms; in Section III we show the results obtained by applying the proposed approaches on a set of hand-labelled videos; finally concluding remarks are given in the last Section.

II. Self-Evaluation of Tracking Algorithm

The proposed online-evaluation method uses motion, shape and appearance features computed at every frame and fed to a Bayesian classifier to obtain a probability of correctness for each tracking decision. The considered features are:

- *Difference of shape ratio between frames*: this score detects rapid changes in the object's shape, which might indicate tracking failure. This value is high if the shape ratio ($R = \frac{W}{H}$, W and H being, respectively, the width and the height of the bounding box containing the object) between consecutive frames $t-1$ and t keeps as constant as possible:

$$\begin{aligned} R_{max} &= max\left\{R_t, R_{t-1}\right\} \\ R_{min} &= min\left\{R_t, R_{t-1}\right\} \\ shape\_ratio\_score &= \frac{R_{min}}{R_{max}} \end{aligned}$$

- *Difference of area between frames*: similarly to the previous score, this value indicates whether the area of the tracked object has a sudden change between two consecutive frames. It is computed in the same way as the shape ratio difference score.

$$\begin{aligned} A_{max} &= max\left\{A_t, A_{t-1}\right\} \\ A_{min} &= min\left\{A_t, A_{t-1}\right\} \\ area\_ratio\_score &= \frac{A_{min}}{A_{max}} \end{aligned}$$

- *Histogram difference*: this feature evaluates the difference between two appearances of the same object by comparing the respective histograms (analyzing independently the three RGB channels and the grayscale versions of the two objects). Given histograms H_t and H_{t-1}, the corresponding score is computed as:

$$\sum_{i=0}^{255} \frac{min\left\{H_t\left(i\right), H_{t-1}\left(i\right)\right\}}{max\left\{H_t\left(i\right), H_{t-1}\left(i\right)\right\}}$$

- *Direction smoothness*: assuming a trajectory is as good as it is regular and without sudden direction changes, this value keeps track of the direction of the object in the last frames and checks for unlikely changes in the trajectory. It is computed as:

$$direction\_smoothness = \frac{|\theta_1 - \theta_2|}{180}$$

where θ_1 and θ_2 are the angles (with respect to the x axis) of the last two displacements of the object. For simplicity, we use $\theta_1 - \theta_2$ in the formula, although the actual implementation handles the case of angles around the $0°/360°$ boundary.

- *Speed smoothness*: similarly to the previous feature, this value checks whether the current speed of the object (i.e. the displacement between the previous position and current one) is similar to the average speed in the object's history. Let P_t and P_{t-1} be the last two positions of the object, we compute $s_t = ||P_t - P_{t-1}||$, so that s_t represents the last displacement (speed) of the object, and compare it with the average speed $\bar{s}$ in order to compute $speed\_smoothness$ as:

$$\begin{aligned} s_{max} &= max\left\{s_t, \bar{s}\right\} \\ s_{min} &= min\left\{s_t, \bar{s}\right\} \\ speed\_smoothness &= \frac{s_{min}}{s_{max}} \end{aligned}$$

- *Texture difference*: mean and variance of Gabor filters at different scales (2, 4, 8, 16) and orientations (0°, 45°, 90°, 135°) are computed from two consecutive appearances and compared. Given two feature vectors v_1 and v_2, this value is computed as the Euclidean norm between the two vectors:

$$\sqrt{\sum_{i=1}^{n} \left(v_1\left(i\right) - v_2\left(i\right)\right)^2} \tag{1}$$

The vector made up of these values for each tracking decision is then given as input to a naive Bayes classifier, which computes the probability of the considered tracking decision being good. Naive Bayes classifier use Bayes theorem to estimate the posterior probability that a feature vector belongs to a certain class, given the estimated distributions (typically, as in this case, assumed Gaussians) of each feature, for that certain class.
For our purpose, we define two classes *"good tracking" (GT)* and *"bad tracking" (BT)* describing a tracking decision whose motion/appearance/shape properties are more likely to derive, respectively, from a correct or a wrong association by the tracker. After training the Bayes classifier on these two classes, the evaluation process consists in computing the above-described feature vector at each tracking decision, feeding it to the classifier and then reading the matching probability between the vector and the *GT* class; this value is then returned as the performance score for that tracking decision.

III. Experimental results

The video base used to test the performance of our on-line evaluation method consisted of 15 videos (30 fps, spatial resolution 320 × 240, 24-bit color depth), depicting three main targets:

- 5 videos from Caltrans Live Traffic Cameras[1]: the main targets were cars, trucks and motorcycles whose motion was constrained to the lanes of a highway.
- 5 videos from the CAVIAR dataset[2] showing people walking in closed environments (e.g. shopping centre);
- 5 underwater videos from the Fish4Knowledge[3] project's dataset. The recorded scenes depict fish swimming in unconstrained real-life environments.

The choice of the different application domains was motivated by the need to train the classifier on as many different scenarios as possible, in order to avoid its performance to be biased by the targets' motion patterns. The ground truths for the CAVIAR videos can be found on the project's website, and it includes information on the position, orientation, bounding box and behaviour hypotheses of the targets. The ground truth for the underwater and vehicular traffic videos were hand-labeled by us using the Video Performance Evaluation Resource (VIPER) [12], and they include information on the

[1] http://video.dot.ca.gov/

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

[3] http://fish4knowledge.eu

bounding box and the contour of the targets. However, for our purposes, the bounding box is the only information we used, since it allowed us to compute all the above-described features.

The Bayes classifier was trained with data coming from 9 videos of our video base (3 for each category). The samples related to the "Good Tracking" *GT* class were generated by computing the feature vectors on correct tracking decisions taken from our ground truth dataset, whereas samples for the "Bad Tracking" *BT* class were artificially generated by either making mis-associations between regions in consecutive frames belonging to different objects, or by randomly translating and modifying the correct object region.

The test phase was performed on the 6 remaining videos (2 for each category) and was meant to assess how the proposed method is able to reflect: 1) errors in tracking decisions by applying it to our ground truth data where error was artificially introduced from 10% to 50%, 2) the performance of tracking algorithms by assessing the quality of the tracks computed by three state of the art algorithms. Moreover, the performance of our method were compared with the ones obtained when the features, described in the previous section, were combined through a weighted mean (as performed by most of the existing approaches [8], [10]) and the ones achieved by the method proposed by Erdem *et al.* in [9]. Since the last two approaches are domain-dependent we set their parameters in order to achieve the best performance for each target. Fig. 1 shows this comparison in terms of average evaluation score achieved for each target (vehicle, people and fish) when the three methods were applied to our ground truth data as the tracking error varies. These results reflect that: 1) the evaluation scores of the proposed method are the highest when using ground-truth tracking information, with a slight decrease from the most constrained environments (e.g. vehicles in a highway) to the least constrained ones (e.g. fish in real-life environments) and 2) when adding tracking noise, the results lower sensibly, although our method tends to reflect better the tracking errors showing an almost linear behavior.

To test the reliability of our method in assessing the performance of tracking algorithms, we performed tracking on ground truth objects (on the same video set) with three state-of-the-art algorithms: CONDENSATION [13], CAMSHIFT [14], and covariance-based tracking [15]. Table I shows the performance of these three algorithms when compared against the ground truth on the 16 videos in terms of *Correct decision rate (CDR)*[4] normalized between 0 and 1 and the average evaluation scores achieved, respectively, by the proposed approach (AES_{BC}), the approach that uses the weighted mean (AES_{WM}) of the features instead of the Bayes classifier and the Erdem *et al.* approach (AES_E).

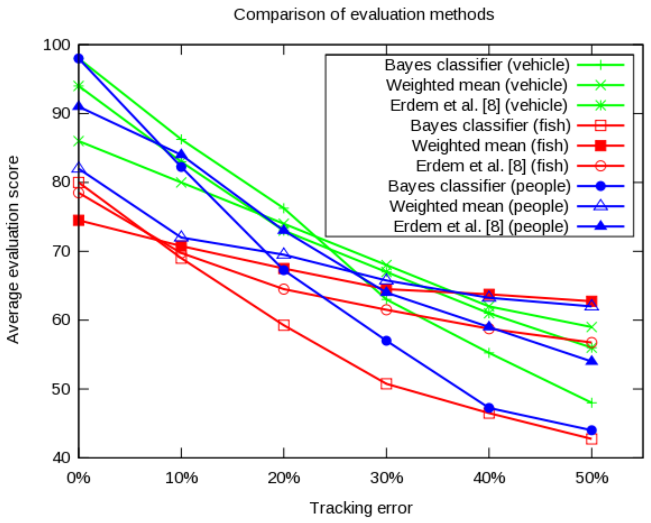
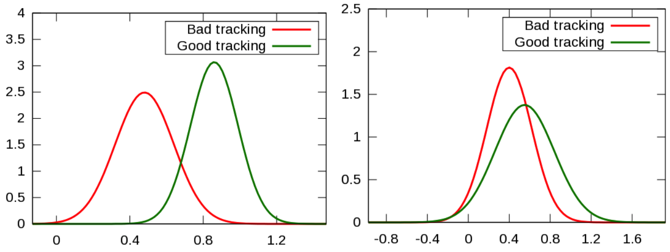


Figure 1. Comparison of the proposed in terms of average evaluation score when the three methods were applied to the ground truth data at varying of tracking errors.



(a) $p\,(area\_ratio|C)$ (b) $p\,(direction\_smoothness|C)$

Figure 2. Distributions of the features $area\_ratio$ and $direction\_smoothness$ in the two classes *GT* and *BT*

The last evaluation aimed at understanding the contribution of each feature to the tracking evaluation score. The use of the Bayes classifier allows us to have a thorough understanding of the features that influence the final evaluation, unlikely existing approaches where the features to be used are identified empirically. Figure 2 shows how the features $area\_ratio$ and $direction\_smoothness$ are distributed in each of the two classes (i.e. the $p\,(area\_ratio|C)$ and $p\,(direction\_smoothness|C)$ distributions, where C is the class label – either *GT* or *BT*).

As it is possible to notice the $direction\_smoothness$ feature showed an overlap between the classes *GT* and *BT*, thus indicating that it does not provide useful information for discriminating a good tracking decision from a bad one, whereas $area\_ratio$ is able to distinguish among the two classes. This holds for all the motion-based features, i.e. $direction\_smoothness$ and $speed\_smoothness$ and Fig. 3 shows that the results in terms of average evaluation score changed slightly (showing very similar trends) when the motion-based features were not used.

[4] Let a "tracking decision" be an association between an object at frame t_1 and an object at frame t_2, where $t_1 < t_2$; such tracking decision is correct if it corresponds to the actual association, as provided by the ground truth. The correct decision rate (CDR) is the percentage of correct tracking decisions of a tracking algorithm when compared with ground truth tracks.

	Vehicles				Human				Fish			
	CDR	AES$_{BC}$	AES$_{WM}$	AES$_{E}$	CDR	AES$_{BC}$	AES$_{WM}$	AES$_{E}$	CDR	AES$_{BC}$	AES$_{WM}$	AES$_{E}$
CONDENSATION [13]	99.1	99.0	86.8	93.2	97.8	97.4	84.8	89.8	94.5	80.2	73.3	79.4
CAMSHIFT [14]	98.8	99.0	86.7	93.2	95.2	96.3	85.1	90.1	91.7	76.3	71.4	75.2
Covariance-based [15]	99.6	99.1	87.2	93.9	98.0	97.4	84.2	90.3	96.7	80.1	74.8	77.3

Table I

TRACKING RESULTS ON THREE DIFFERENT TARGETS COMPARED WITH THE AVERAGE EVALUATION SCORES ACHIEVED, RESPECTIVELY, BY THE PROPOSED APPROACH (AES_{BC}), THE APPROACH THAT USES THE WEIGHTED MEAN (AES_{WM}) OF THE FEATURES INSTEAD OF THE BAYES CLASSIFIER AND THE ERDEM *et al.* APPROACH (AES_{E})

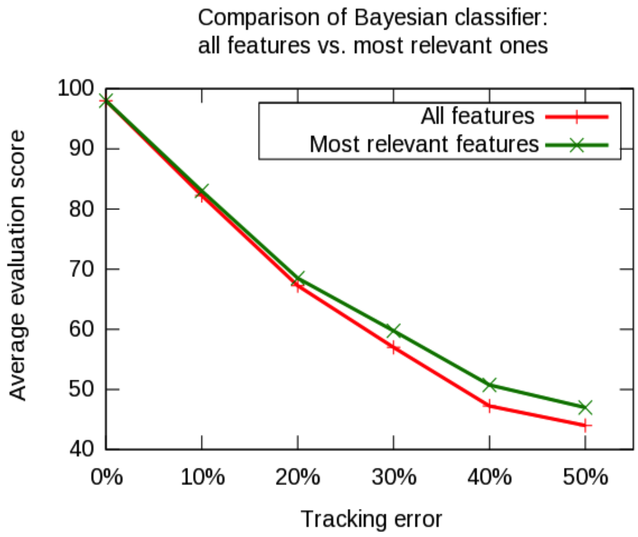


Figure 3. Comparison in terms of average evaluation score when we used, respectively, all the features and only the most relevant ones.

IV. CONCLUDING REMARKS

Object tracking is one of the most important processing blocks in computer vision systems. In recent years, we have been assisting to a proliferation of tracking algorithms; however one of the main problems, still unsolved, in object tracking research is the performance evaluation, which is a difficult and tedious task especially if it is based on hand-labeled ground truth data. In order to address this need, in this paper we have proposed a probabilistic self-evaluation approach that gives a score to each tracking decision. The experimental results have shown how the proposed method is able to reflect the quality of tracking decisions and the performance of different tracking algorithms also under extreme conditions such as in underwater scenes. Moreover, the results have also shown that shape and appearance features (differently from the motion-based ones) are discriminant of good and bad tracking decisions. As future work, we are planning to develop a similar approach to test performance of object detection algorithms and to publish the systems on a web-platform in order to allow researchers to test and compare their algorithms' performance and to share the results with the whole community.

V. ACKNOWLEDGEMENTS

This research was funded by European Commission FP7 grant 257024, in the Fish4Knowledge project (www.fish4knowledge.eu).

REFERENCES

[1] C. Spampinato, Y. H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, tracking and counting fish in low quality unconstrained underwater videos," in *3rd International Conference on Computer Vision Theory and Applications (VIS-APP08)*, 2008, p. 514519.

[2] A. Faro, D. Giordano, and C. Spampinato, "Evaluation of the traffic parameters in a metropolitan area by fusing visual perceptions and cnn processing of webcam images," *IEEE Transactions on Neural Networks*, vol. 19, no. 6, pp. 1108–1129, 2008.

[3] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *Proceedings of ARTEMIS '10*, 2010, pp. 45–50.

[4] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[5] N. Gordon, A. Doucet, and N. Freitas, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, 1979.

[6] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, aug. 2011.

[7] X. Sun, H. Yao, and S. Zhang, "A novel supervised level set method for non-rigid object tracking," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, pp. 3393 –3400.

[8] D. P. Chau, F. Bremond, and M. Thonnat, "Online evaluation of tracking algorithm performance," in *The 3rd International Conference on Imaging for Crime Detection and Prevention*, December 2009.

[9] C. Erdem, A. M. Tekalp, and B. Sankur, "Metrics for performance evaluation of video object segmentation and tracking without ground truth," *Proceedings of Internation Conference on Image Processing*, vol. 2, pp. 69–72, 2001.

[10] H. Wu and Q. Zheng, "Self-evaluation for video tracking systems," Maryland Univ. College Park, Dept. of Eletrical and Computer Engineering, Tech. Rep., 2004.

[11] H. Wu, A. Sankaranarayanan, and R. Chellappa, "Online empirical evaluation of tracking algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 8, pp. 1443–1458, Aug. 2010.

[12] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4, 2000, pp. 167 –170 vol.4.

[13] M. Isard and A. Blake, *International Journal of Computer Vision*, vol. 29, no. 1, 1998.

[14] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.

[15] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.