Symbiotic Tracker Ensemble With Feedback Learning

Victor H. A. Quirita[*], Patrick N. Happ[*], Gilson A. O. P. Costa[†] and Raul Q. Feitosa[*†]
[*]Electrical Engineering Department, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro 22430-060, Brazil
Email: {vhaymaq, patrick, raul}@ele.puc-rio.br
[†]Informatics and Computer Science Department, State University of Rio de Janeiro, Rio de Janeiro 20550-900, Brazil
Email: gilson.costa@ime.uerj.br

Abstract—**Visual tracking is a challenging task due to a number of factors, such as occlusions, deformations, illumination variations and abrupt motion changes present in a video sequence. Generally, trackers are robust to some of these factors, but do not achieve satisfactory results when dealing with multiple factors at the same time. More robust results when multiple factors are present can be obtained by combining the results of different trackers. In this paper we propose a multiple tracker fusion method, named Symbiotic Tracker Ensemble with Feedback Learning (SymTE-FL), which combines the results of a set of trackers to produce a unified tracking estimate. The novelty of the method consists in providing feedback to the individual trackers, so that they can correct their own estimates, thus improving overall tracking accuracy. The proposal is validated by experiments conducted upon a publicly available database. The results show that the proposed method delivered in average more accurate tracking estimates than those obtained with individual trackers running independently and with the original approach.**

Index Terms—**Object tracking, tracking fusion.**

I. Introduction

The basic task of visual tracking is to estimate the state (position and extent) of arbitrary objects along image sequences. Many automated applications such as video surveillance, human-computer interaction, and smart traffic monitoring, rely on the information delivered by a tracker in order to help machines to understand real-world environment, recognize object behavior and take particular actions when necessary. The target objects are, however, prone to changes in appearance due many factors such as occlusions, deformations, illumination variations and abrupt motion changes, which make visual tracking a challenging problem in real-world scenarios.

Over the past two decades, several tracking algorithms have been proposed in the pursuit of accuracy and robustness (e.g. [1]–[4]) but with limited success in the presence of the different aforementioned variations [5]. In fact, a tracker may be designed for dealing with specific situations, but would fail when facing different conditions [6]. In this sense, fusion techniques have been emerging as a way to improve overall tracking performance, as they have the capacity to exploit complementary information among trackers aiming at more accurate results [7]. However, most of the available fusion techniques limit themselves to provide more accurate consensus outcomes and do not exploit this result to improve individual trackers' performances.

In this context, we propose a multiple tracker fusion approach, which exploits the fusion results in order to enhance the performance of the individual trackers. The fusion is expected to provide a more consistent result by resubmitting its results to each tracker in the attempt of improving their performances. The proposed method is based on the Symbiotic Tracker Ensemble introduced by Gao et al. [8], which considers trackers as black-boxes and combines their results based on intra-tracker and inter-tracker correlation. The main contribution in this work is the inclusion of a feedback mechanism, which updates object representation by correcting the state of the target object or by updating the appearance model of each individual tracker.

The rest of this paper is organized as follows. Section II reviews previous methods for the fusion of trackers' estimates. Section III describes the fundamentals of Gao's Symbiotic Tracker Ensemble [8]. Section IV introduces the methodology adopted to include the feedback mechanism into the fusion approach in order to enhance the individual trackers' performances. In Section V, we briefly describe the three state-of-the-art tracking algorithms used in this work, namely, Tracking-Learning-Detection (TLD) [9], Kernelized Correlation Filters (KCF) [10] and Circulant Structure of Tracking-by-Detection with Kernels (CSK) [11], and we discuss the results obtained with the individual trackers and with the fusion approaches on the TB-50 dataset [12], and finally, Section VI summarizes our conclusions and suggestions for future work.

II. Related work

The massive volume of information currently produced by a multitude of video sources, including cameras and built in mobile devices, enable many important applications in fields as diverse as security, health, sports, transportation, and so on. Such wealth of information exposes an unquestionable demand for robust automatic visual tracking methods.

Although exhaustive research has been made in order to produce accurate and robust tracking algorithms [13]–[15], the task of visual tracking remains challenging due to the presence of perturbations during image acquisition, such as deformations, illumination changes, occlusions, cluttered scenes, low frame rate, abrupt motion, etc. The works of Wu [5], Smeulders [6] and Kristan [16] reveal that individual trackers cannot cope with all kinds of perturbations; furthermore, the performance of the evaluated trackers decreases when simultaneous perturbations occur. However, their results also suggest that trackers might complement each other, as some of trackers perform well in situations where others perform poorly.

Several authors have proposed fusion techniques in an attempt to improve tracking performance. Shearer et al. [17] proposed a method that allows switching between estimates of two trackers: a region tracker and an edge tracker, according to a confidence measure, but it requires user intervention when possible drifts are detected. Leichter and co-workers [18] devised a method that combines several tracking estimates through the exchange of their final state *pdf* (probability density function); the method is, however, limited to trackers of the same nature. As a way to combine trackers in a more general framework, Stenger et al. [19] proposed to select the best suited trackers for a given application, based on error distributions learned from a representative training set. Stenger's approach is, nevertheless, limited to a certain number of trackers and to a range of perturbations present solely during training. The disagreement-based fusion approach proposed by Li et al. [20] have similar restrictions with respect to the number and type of trackers. In a limited, yet interesting fusion approach, Bailer et al. [21] combined the estimates of a set of trackers through a trajectory optimization scheme, where tracking results for a given video sequence were known in advance.

The method introduced in this paper is based on the symbiotic tracker ensemble proposed by Gao et al. [8], which is a fusion algorithm that depends only on the trackers' estimates regardless of their design, i.e., treating trackers as black-boxes. Although Gao's approach is fairly general, it does not support updating the object's representation of the individual trackers. In this sense, Leang et al. [22] evaluated different strategies for updating or re-initializing trackers by combining fusion outputs and drift predictions. However, each tracker's contribution is given by a binary confidence level, which considers tracker's performance from the previous and current frames instead of the accumulated performance during tracking. Zhong et al. [23] also proposed that the fusion of tracking estimates can be used to update individual trackers in order to improve their accuracy. Finally, Biresaw et al. [24] proposed a fusion framework that enables individual tracker correction based on estimates provided by other trackers, but the method is restricted to Bayesian trackers.

III. Original Symbiotic Tracker Ensemble

The Symbiotic Tracker Ensemble [8] is a fusion framework that explores the relationships among the estimates from a set of trackers regardless their particular designs, i.e., trackers are treated as black-boxes.

The fusion process considers spatiotemporal characteristics of the individual trackers' outcomes (Figure 1) in two iterative stages. First, intra-tracker correlation is used to evaluate the tracking consistency, i.e., trajectory smoothness between tracking estimates for consecutive frames. Then, inter-tracker correlation is used to calculate the confidence level of each tracker, through a pair-wise tracker interaction. In the following we describe a set of relationships between tracking estimates, and we give a brief explanation of intra-tracker and inter-tracker correlations, and of the computation of the final estimate.

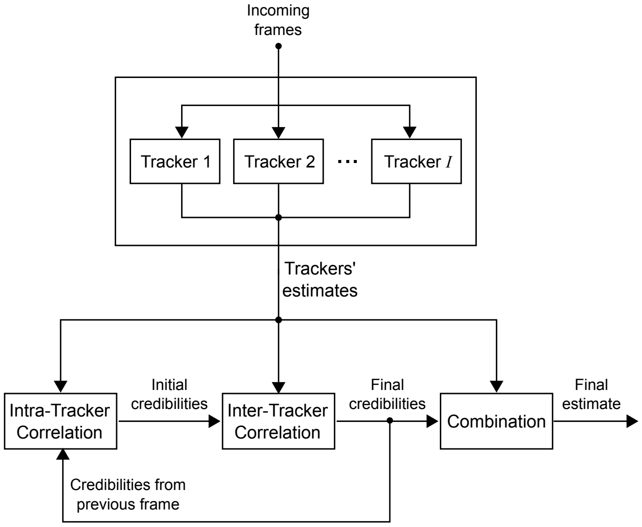


Fig. 1. Symbiotic Tracker Ensemble.

A. Relationships Between Tracking Estimates

A key problem in designing a fusion method for tracking is to define how to express numerically the relationships between two tracking estimates, R_1 and R_2, which can be represented by bounding boxes in the form $R = (x, y, width, height)$. For this purpose, Gao et al. [8] introduce two similarity metrics, formally:

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

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

  where $F(R_1, R_2) \in [0, 1]$, and $Pr(R_1, R_2)$ and $Re(R_1, R_2)$ represent precision and recall, respectively.
- $r(R_1, R_2)$: which quantifies the congruence between R_1 and R_2 according to:

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

  where $r(R_1, R_2) \in [0, 1]$, $D(R_1, R_2)$ represents the Euclidean distance between the centers of R_1 and R_2, and σ stands for the standard deviation of $D(R_1, R_2)$.

B. Intra-Tracker Correlation

The first stage of the fusion approach evaluates individual tracker consistency. For each tracker, estimates from successive frames are used to compute a temporal correlation measure, which determines its initial credibility. In a more formal way, given two tracking estimates $R_{i,n-1}$ and $R_{i,n}$, corresponding to the i-th tracker at $(n-1)$-th and n-th frames, the initial credibility is defined by:

$$C_{i,n}^0 = \xi_i C_i + (1 - \xi_i)\Theta(R_{i,n-1}, R_{i,n})C_{i,n-1}^f \quad (3)$$

where, C_i is a general confidence coefficient, $C_{i,n-1}^f$ representing the final credibility coefficient from previous frame, ξ_i is a regularization parameter that ranges between 0 and 1, and $\Theta(\cdot)$ is a relation coefficient of the i-th tracker that may be computed using either $F(\cdot)$ or $r(\cdot)$ similarity metric.

C. Inter-Tracker Correlation

The second stage of the fusion approach computes individual trackers' confidences by comparing all trackers' outputs for a single frame. The individual credibility is estimated trough an iterative pair-wise correlation, as presented in Equation 4, where $C_{i,n}^{s-1}$ represents the credibility coefficient for the i-th tracker after the s iteration, $\eta_i \in [0, 1]$ is a weighting coefficient that controls the importance of the temporal correlation, I denotes the total number of trackers, $R_{j,n}$ and $R_{i,n}$ are the tracking estimates for the i-th and j-th trackers, respectively, and $\Phi(\cdot)$ is a relation coefficient between the i-th and j-th trackers, which may be computed using either $F(\cdot)$ or $r(\cdot)$ metric.

$$C_{i,n}^s = \eta_i C_{i,n}^0 + \frac{1 - \eta_i}{I - 1} \sum_{i \neq j} \Phi(R_{j,n}, R_{i,n})C_{i,n}^{s-1} \quad (4)$$

Notice that after convergence the credibility coefficients $C_{i,n}^s$ becomes the final credibility coefficients $C_{i,n}^f$.

D. Estimates Combination

The last stage in the fusion approach computes the final estimate through a weighted sum of the trackers' outputs, formally:

$$R_{fusion} = \sum_i \pi_i R_i \quad (5)$$

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

$$\pi_i = \frac{C_{i,n}^f}{\sum_j C_{j,n}^f} \quad (6)$$

IV. Symbiotic Tracker Ensemble with Feedback Learning

The symbiotic tracker ensemble, as described in the previous section, is based on a spatiotemporal correlation of trackers' estimates, and is fairly robust to different perturbations in a video sequence. However, it does not address directly two important issues that might compromise performance:

1) *Drifting*, which is a consequence of the independent execution of the trackers, as they are prone to add information into the representation of the object that does not correspond with the target.
2) *Perspective bias*, which is a consequence of considering only the centroids to compare two tracking estimates disregarding their sizes [25].

In this work, we attempt to overcome the first issue by assuming that the fusion estimate tends to correspond to the real state of the object. Thus, we can use this information to correct eventual drifts of individual trackers. Therefore, we included a feedback mechanism into the original fusion scheme, as depicted in Figure 2, which is used to correct the state of the object and, if applicable, update the appearance model of the trackers that take part in the ensemble.

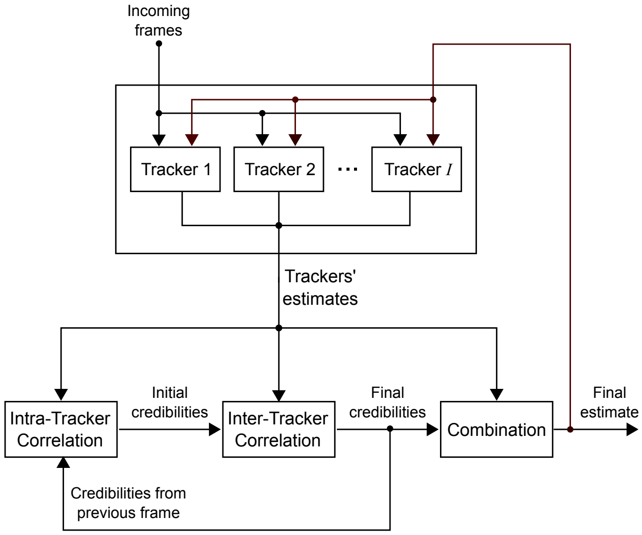


Fig. 2. Symbiotic Tracker Ensemble with Feedback Learning.

Additionally, we take advantage of the structure of the tracking algorithms – first estimate the object's state and then update the object's appearance model – to keep the trackers updated with information delivered by the fusion output. The basic processing steps of the proposed approach is presented in Algorithm 1. They consist of four steps: execution of the individual trackers; fusion of trackers' results; correction of trackers' states; and update trackers' appearances models.

Notice that the proposed method remains general in the sense that any tracker, whether adaptive [9] or tracking-by-detection [26], can be used in the fusion scheme. A small modification of the trackers' code is, however, required to enable the feedback process.

Finally, we use a normalized version of the Euclidean Distance to redefine $r(\cdot)$ in Equation 2 in order to attenuate the *perspective bias* effect. Formally, given two tracking estimates in the form of a bounding box, $R_1 = (x_1, y_1, width_1, height_1)$ and $R_2 = (x_2, y_2, width_2, height_2)$, and the centroids (u_1, v_1) and (u_2, v_2) corresponding to R_1 and R_2, respectively, the Normalized Euclidean Distance (NED) is defined as:

$$NED(R_1, R_2) = \sqrt{d^2(u_1, u_2) + d^2(v_1, v_2)} \quad (7)$$

where,

$$d(u_1, u_2) = \frac{u_1 - u_2}{width_1} \quad (8)$$

and,

$$d(v_1, v_2) = \frac{v_1 - v_2}{height_1} \quad (9)$$

Algorithm 1 Symbiotic Tracker Ensemble with Feedback Learning

```
Input:
$frames$ - sequence of images.
$T \leftarrow \{T_i\}$ - tracker ensemble.
$R_0$ - initial state.
$C, \xi, \eta$ - fusion parameters.
$\theta$, $\phi$ - relationships' estimates in $\Theta(\cdot)$ and $\Phi(\cdot)$, respectively.

Procedure:
$n \leftarrow 1$;
$frame \leftarrow$ GetFrame($frames$, $n$);
InitializeTrackers($T$, $R_0$, $frame$);
InitializeFusion($C$, $\xi$, $\eta$);
for each $frame$ $n$ from $frames$ do
    $frame \leftarrow$ GetFrame($frames$, $n$);
    for each $tracker$ $i$ from $T$ do
        $R_i \leftarrow$ ExecuteTracker($T_i$, $frame$);
        $C^0_{i,n} \leftarrow$ ComputeIntraTrackerCorrelation($R_i$, $C^f_{i,n-1}$, $\theta$);      ▷ Equation 3
        $C^f_{i,n} \leftarrow$ ComputeInterTrackerCorrelation($\{R_i\}$, $C^0_{i,n}$, $\phi$);      ▷ Equation 4
    end for
    $R_{fusion} \leftarrow$ CombineTrackingEstimates($\{R_i\}$, $\{C^f_{i,n}\}$);      ▷ Equation 6
    CorrectTrackersStates($T$, $R_{fusion}$);
    UpdateTrackersAppearances($T$, $frame$, $R_{fusion}$);
end for
```

In this way, we redefine $r(\cdot)$ as:

$$r_{NED}(R_1, R_2) = exp(-\frac{NED^2(R_1, R_2)}{\sigma^2}) \tag{10}$$

V. Experimental Design and Results

In this section, we evaluate the proposed method, the Symbiotic Tracker Ensemble with Feedback Learning (SymTE-FL), using a collection of video sequences publicly available. Three state-of-the-art tracking-by-detection algorithms: Tracking-Learning-Detection (TLD) [9]; Kernelized Correlation Filters (KCF) [10]; and Circulant Structure of Tracking-by-Detection with Kernels (CSK) [11], compose the ensemble. We compare the results delivered by the proposed method with those achieved by the original Symbiotic Tracker Ensemble (SymTE) [8] and by each individual tracker running independently, i.e. without the feedback learning. The experiments were carried out on an Intel(R) Core(TM) i7-3930K, 3.20GHz CPU with 32GB of RAM running Windows 7, and implemented with MATLAB R2016a.

We evaluate tracking performance by first computing the errors in precision and distance associated to each tracker estimates, and by analyzing the areas under the curves of precision plots, as explained later in this section.

A. Dataset

In the experiments we used the TB-50[1] dataset, which is a collection of difficult and representative video sequences commonly used in tracking evaluations [5]. The dataset contains 50 sequences in which the targets are subjected to challenging situations like illumination variations, occlusions, deformations, motion blurriness, in-plane and out-of-plane rotations, background clutters, and low resolutions. The sequences have approximately 71∼1918 frames; the resolution of the frames varies between (128∼768)×(96∼640) pixels; and targets' initial widths and heights vary from (12∼132) to (13∼210) pixels.

B. Trackers

As mentioned before, we used three state-of-the-art tracking-by-detection algorithms: TLD, KCF and CSK, which codes are publicly available. We have adjusted the trackers' codes as mentioned in Section IV, and used predefined parameter settings for each tracker. Additionally, we modified the KCF and CSK algorithms to make them capable of handling scale variations of the targets.

1) Tracking-Learning-Detection (TLD): TLD [9] is a framework conceived to perform long-term tracking of arbitrary objects; it combines adaptive tracking and online detection techniques in order to acquire and exploit temporal information about the target and, thus, overcome possible appearance changes of the object during tracking. As the name suggests, TLD decomposes long-term tracking into three stages, namely:

- *Tracking*: estimates the object's state in incoming frames.
- *Learning*: analyzes the responses from the tracking and detection stages in order to identify detection errors and generate reliable data for training.

[1] Available in: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html. Last accessed in June 2017.

- *Detection*: locates the object, either to correct tracking trajectory or to re-start tracking after failure.

2) Circulant Structure of Tracking-by-Detection with Kernels (CSK): CSK [11] is a tracker-by-detection algorithm that uses a non-linear mapping process to discriminate potential image patches as the foreground (object's appearance model) or the background. The classifier is trained during tracking execution using a set of samples collected within an area near to the last estimates of the target's position. Image patches relatively close to the target's estimates are labeled as positive samples, while those farther away are considered negative ones.

3) Kernelized Correlation Filters (KCF): KCF [10] is a tracking algorithm built on the CSK tracker basic design. Differently from CSK, this method extends the use of Gaussian kernels to linear and polynomial kernels. Furthermore, the target object is represented by more complex features, such as the histogram of gradients, in a multi-channel image representation. Finally, KCF uses the whole frame extent to extract image patches in order to feed the classifier with positive and negative samples.

C. Fusion Parameters

The use of $\Theta(\cdot)$ and $\Phi(\cdot)$ in Equations 3 and 4, respectively, enables four variants of the proposed method (SymTE-FL), as well as four variants of the SymTE method.

We identify the variants as: FF, FD, DF and DD. The first letter refers to the similarity metric used to compute intra-tracker correlation ($\Theta(\cdot)$), and the second, to the metric used to compute inter-tracker correlation ($\Phi(\cdot)$). The letter F is used to denote the $F(\cdot)$ metric, and the letter D is used to denote the $r_{NED}(\cdot)$ metric.

For the $r_{NED}(\cdot)$ similarity metric, we set the value of σ equal to $1/3$, which was determined empirically. Additionally, the values of ξ_i and η_i were both set to 0.1 according to the reported values in [8]. Finally, we chose equal confidence coefficients (C_i) for each tracker in the ensemble.

D. Evaluation measures

We used the Normalized Euclidean Distance Error, e_{NED}, and Precision Error, e_P, to evaluate tracking performance. The former measures the deviation of a tracker estimate (R) from the reference (R_{gt}), and the latter measures similarity in position and extent between R and R_{gt}. Formally, e_{NED} and e_P are defined as follows:

$$e_{NED}(R_{gt}, R) = 1 - r_{NED}(R_{gt}, R) \tag{11}$$

$$e_P(R_{gt}, R) = 1 - Pr(R_{gt}, R) \tag{12}$$

where, $r_{NED}(\cdot)$ is the distance-based metric defined in Section IV; and $Pr(\cdot)$ is the precision between a pair of bounding boxes.

We computed the area under the curve (AUC) of the precision curves, for both e_{NED} and e_P, to evaluate tracking performance in a given video sequence. The precision curves are commonly used in literature [3], [5] to represent the proportion of frames in a sequence where the target is assumed to be correctly tracked, i.e., where tracking error is below a given threshold. We observe that a good tracker performance is associated to a high AUC values.

E. Results

Figures 3 and 4 present the average results for the area under the curve of the precision curves related to e_{NED} and e_P, respectively, considering all the sequences in the TB-50 dataset.

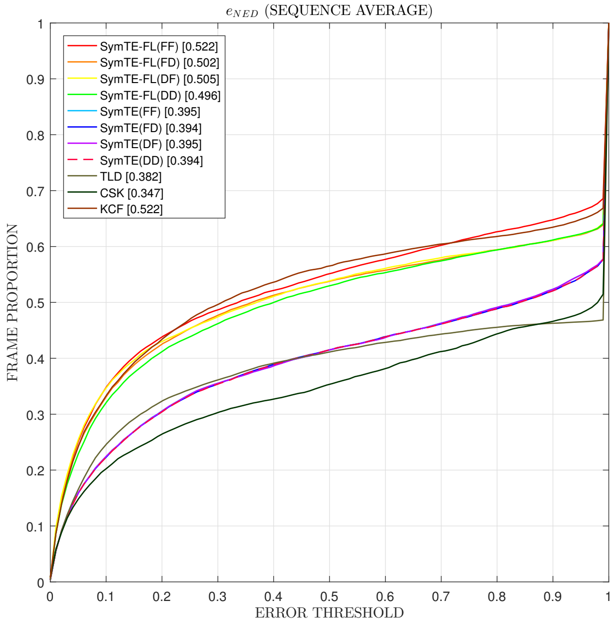


Fig. 3. Precision plot for e_{NED} errors in all the TB-50 sequences.

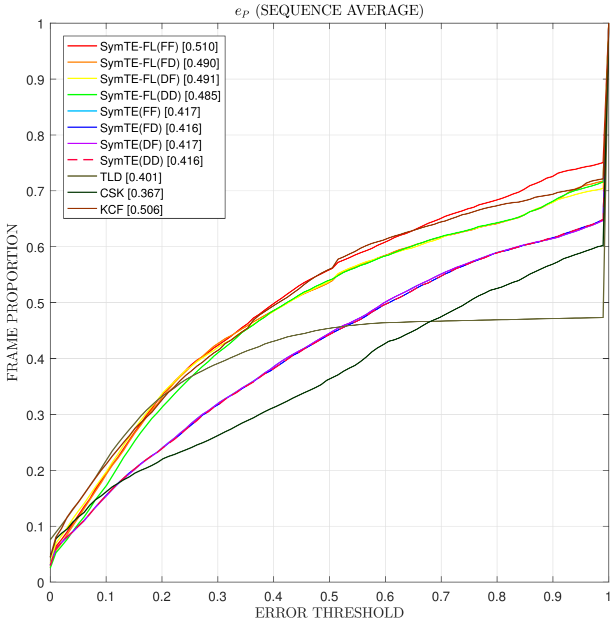
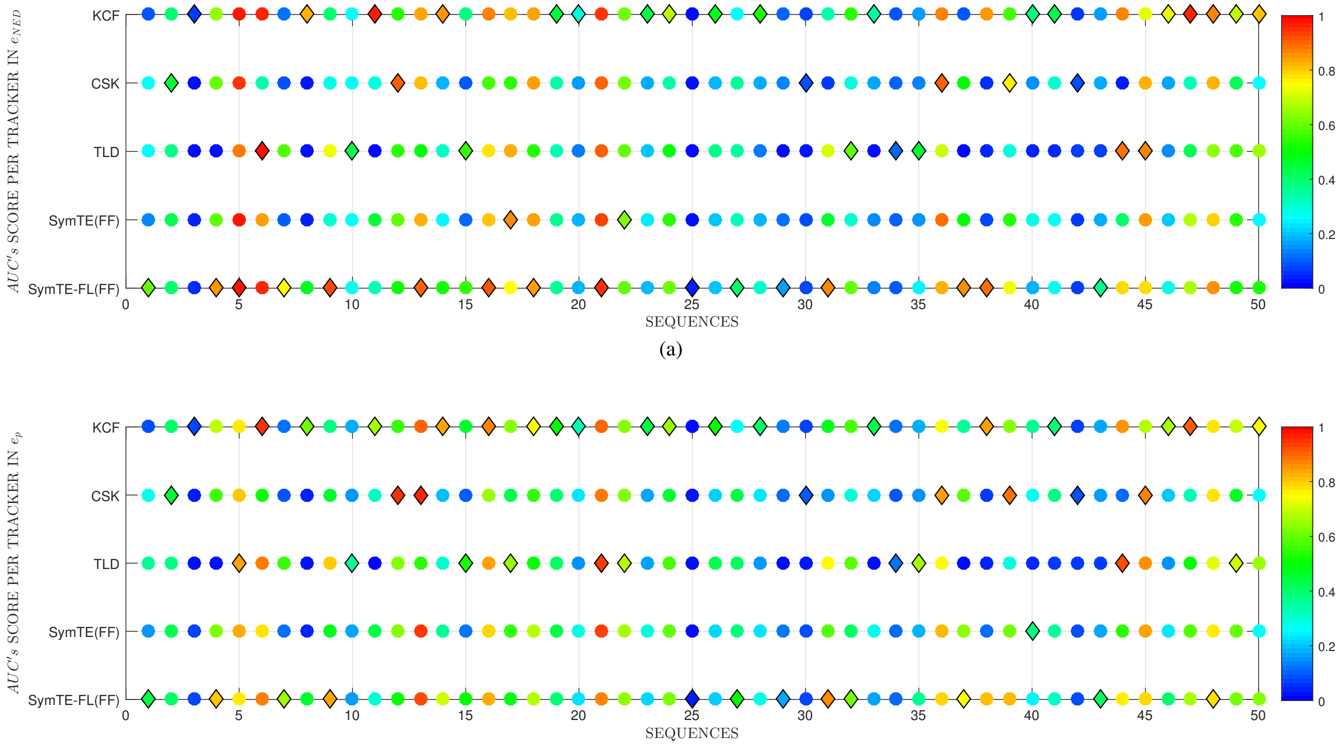


Fig. 4. Precision plot for e_P errors in all the TB-50 sequences.



(a)

(b)

Fig. 5. AUC's ranking for (a) e_{NED} and (b) e_P errors in all the TB-50 sequences. Diamond markers indicate the method with the highest score in the sequence.

These Figures show that, in average, the worse variant of the proposed method (SymTE-FL(DD)) outperforms the best variant of SymTE with respect to both measures, $AUC(e_{NED})$ and $AUC(e_P)$. The performance of the proposed method also surpasses consistently the performances of the TLD and CSK by approximately 10%. In comparison with KCF, the SymTE-FL(DD) obtained similar results for the FF variant and worse results for FD, DF and DD. These facts can be analyzed in more details when looking at Figure 5.

Figure 5 illustrates the trackers' performances, regarding the similarity to the reference, for each sequence. Warmer colors, like red, are associated to better performances in contrast to colder colors, like blue, which show a high estimate deviation from the reference. Additionally, diamond markers indicate the method with the highest score for a given sequence. For the sake of clarity, we did not show FD, DF and DD variants on the graph, since their results are relatively similar to the ones obtained by FF.

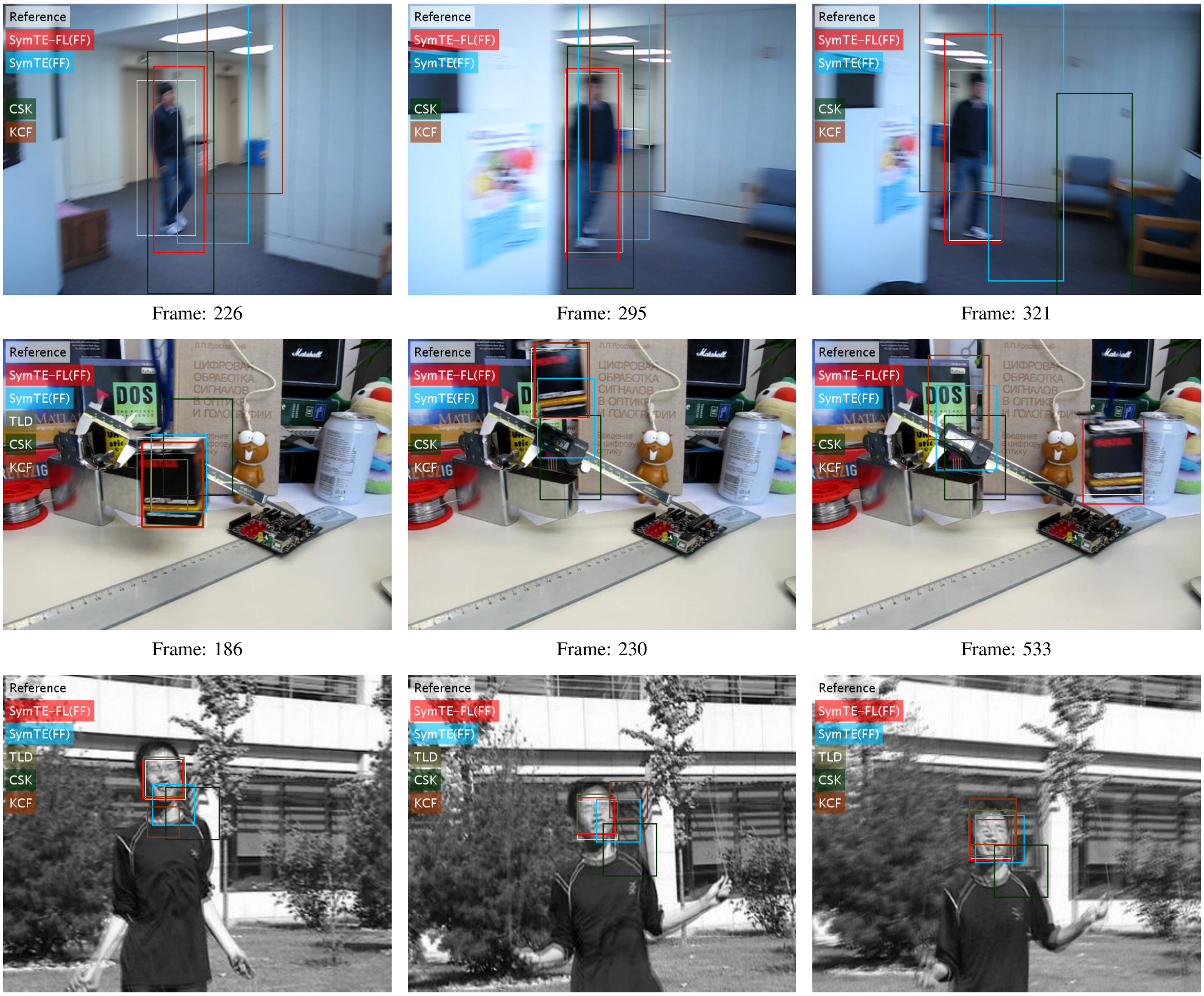
The figure clearly shows that each individual tracker outperforms the others in at least some sequences. KCF is the best one for 18 sequences, while TLD and CSK achieve the best results for 8 and 6 sequences, respectively. The original fusion method, SymTE, appears only 2 times at the first rank, whereas the proposed method, SymTE-FL, 16. These results indicates that KCF was much superior to the other trackers, which made the fusion process to be unbalanced, tending to favor the KCF outcomes. For this reason, SymTE-FL achieved results comparable to those of KCF.

Figure 5 also allows to analyze not only the best trackers, but the difference among them in each sequence. For instance, we can observe that KCF is fairly better for sequences 8, 11 and 47; while CSK is for 12, 37 and 39; and TLD is for 9, 10 and 31. However, there are some sequences in which the trackers have performed quite similar, like 2, 3, 25, 30, 42.

In this sense, Figure 5 shows that, in general, the proposed method produced results similar to the best individual trackers for most of the video sequences. In fact, our approach achieved not only the best average results, but also obtained much higher scores in some individual sequences like in 1, 4, 7, and so on, indicating that the feedback learning tends to increase the individual trackers' performances. However, notice that in some cases the original fusion approach was superior to ours, as in 17, 36 and 45, showing that the feedback is not always advantageous.

In order to present some visual examples, Figure 6 shows the tracking and fusion estimates of three video sequences from the TB-50 dataset. The images on the top, middle, and bottom rows correspond to different frames on BlurBody (sequence 4), Box (sequence 9), and Jumping (sequence 31) sequences, respectively.

The trackers' estimates vary a lot in BlurBody sequence. For instance, KCF is a little bit far from the reference in frame 226 and gets closer to it in frames 295 and 321. In contrast to it, CSK is close to the reference in frames 226 and 295, but drifts away in frame 321, making the SymTe to drift too. In the Box sequence, KCF and TLD are close to the reference and



Frame: 226 Frame: 295 Frame: 321

Frame: 186 Frame: 230 Frame: 533

Frame: 70 Frame: 105 Frame: 274

Fig. 6. Tracking and fusion estimates of three sequences from the TB-50 dataset. Images on the top, middle, and bottom rows present estimates' examples on the BlurBody, Box, and Jumping sequences, respectively.

CSK is slightly drifted in Frame 186. However, as the object moves, the trackers start to lose the correct reference. In frame 533, only our proposed method is tracking correctly the object. Lastly, in Jumping sequence, TLD keeps tracking the object correctly through the frames, but the CSK and KCF drifts away, making the SymTE to drift too, while our proposed method remains very close to the reference. These examples show that, in contrast to the fusion estimate from SymTE, our method manages to keep on following the objects of interest, since the trackers are improved through the feedback learning process.

Finally, the whole experimental results suggests that adding more trackers that are robust to different variations, and that tuning of the fusion parameters should improve the symbiotic tracker ensemble with feedback learning even further. A greater number of different trackers should achieve a more variation on the best trackers for each sequence, thus it would contribute to the fusion method to be more consistent and to obtain, on average, the best performance.

VI. Conclusion

In this paper we proposed a multiple tracker fusion method, the Symbiotic Tracker Ensemble with Feedback Learning (SymTE-FL), which combines the results of the individual trackers to produce a unified tracking estimate. The method was based on a previously proposed fusion method [8] and includes a novel feedback mechanism that corrects the outcomes of the individual trackers that compose the ensemble.

A set of experiments were conducted upon the TB-50 publicly available database. In the experiments we compared the performance of the proposed method with that of the original method, using the same individual trackers: TLD, KCF and CSK. We also compared the performance of the proposed method with those of the individual trackers running independently.

The results show that in general, the proposed method achieved a performance similar, if not better, compared to the best trackers for most evaluated video sequences. In terms of the average area under the curve (AUC) of the precision curves considering the whole dataset, the proposed method outperformed the original method (using the same individual trackers) and the TDL and CSK trackers by approximately 10%. The KCF tracker, however was slightly superior in terms of the AUC, using the normalized Euclidian distance error measure.

We observe that the setup of the proposed method for the experiments gave equal importance to all the trackers in the ensemble, and sometimes a poor tracking estimate produced by a particular tracker may impair the performance of the fusion scheme and even worsen the outcomes of the individual trackers. This fact is more noticeable in the performed experiments due to the small number of trackers (three) that take part in the ensemble. We expect to achieve more accurate and robust results by using a larger number of trackers.

This study is part of an ongoing research, in which we foresee the inclusion of additional mechanisms to improve the tracker ensemble performance. In the near future, we also plan to experiment with different ensembles, using a larger number of different individual trackers, over a variety of video sequence databases.

Acknowledgment

This work was partially supported by CNPq (Conselho Nacional de Desenvolvimento e Pesquisa), CAPES (Coordenação de Aperfeiçoamento de Pessoal) and by FAPERJ (Fundação de Amparo Pesquisa do Estado do Rio de Janeiro).

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.

[2] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, May 2008.

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

[4] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, October 2016.

[5] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, September 2015.

[6] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, July 2014.

[7] J. Kwon and K. M. Lee, "Tracking by sampling and integrating multiple trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1428–1441, November 2013.

[8] Y. Gao, R. Ji, L. Zhang, and A. Hauptmann, "Symbiotic tracker ensemble toward a unified tracking framework trackers–the" black boxes" approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 7, pp. 1122–1131, January 2014.

[9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, July 2012.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, March 2015.

[11] ——, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the 12th European Conference on Computer Vision, Part IV (ECCV'12)*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2012, pp. 702–715.

[12] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*. Washington, DC, ISA: IEEE, June 2013, pp. 2411–2418.

[13] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, September 2009, pp. 1409–1416.

[14] S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: Survey and evaluation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4334–4348, October 2012.

[15] J. Zhang, L. Yang, and X. Wu, "A survey on visual tracking via convolutional neural networks," in *Proceedings of the 2nd IEEE International Conference on Computer and Comunications*, October 2016, pp. 474–479.

[16] M. Kristan, J. Matas, A. Leonardis, T. Voj, R. Pflugfelder, G. Fernndez, G. Nebehay, F. Porikli, and L. ehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137–2155, November 2016.

[17] K. Shearer, K. D. Wong, and S. Venkatesh, "Combining multiple tracking algorithms for improved general performance," *Pattern Recognition*, vol. 34, no. 6, pp. 1257–1269, June 2001.

[18] I. Leichter, M. Lindenbaum, and E. Rivlin, "A general framework for combining visual trackers–the" black boxes" approach," *International Journal of Computer Vision*, vol. 67, no. 3, pp. 343–363, May 2006.

[19] B. Stenger, T. Woodley, and R. Cipolla, "Learning to track with multiple observers," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. IEEE, August 2009, pp. 2647–2654.

[20] Q. Li, X. Wang, W. Wang, Y. Jiang, Z.-H. Zhou, and Z. Tu, "Disagreement–based multi–system tracking," in *Proceedings of the 11th International Conference on Computer Vision - Volume 2*, ser. ACCV'12. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 320–334.

[21] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *Proceedings of the 13th European Conference on Computer Vision - Part VII*, ser. Lecture Notes in Computer Science. Cham: Springer International Publishing, September 2014, pp. 170–185.

[22] I. Leang, S. Herbin, B. Girard, and J. Droulez, "Robust fusion of trackers using online drift prediction," in *Proceeding of the 16th International Conference on Advanced Concepts for Intelligent Vision Systems*. Cham: Springer International Publishing, October 2015, pp. 229–240.

[23] B. Zhong, H. Yao, S. Chen, R. Ji, T.-J. Chin, and H. Wang, "Visual tracking via weakly supervised learning from multiple imperfect oracles," *Pattern Recognition*, vol. 47, no. 2, pp. 1395–1410, March 2014.

[24] T. Biresaw, A. Cavallaro, and C. Regazzoni, "Tracker-level fusion for robust bayesian visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 776–789, May 2015.

[25] E. Maggio and A. Cavallaro, *Video Tracking: Theory and Practice*, 1st ed. John Wiley & Sons, February 2011.

[26] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, November 2011.