Coherent Selection of Independent Trackers for Real-time Object Tracking

Salma Moujtahid, Stefan Duffner, Atilla Baskurt

Coherent Selection of Independent Trackers for Real-time Object Tracking

Salma Moujtahid, Stefan Duffner and Atilla Baskurt
Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, Villeurbanne, France
{*salma.moujtahid, stefan.duffner, atilla.baskurt*}*@liris.cnrs.fr*

Keywords: Visual Object Tracking, Classifier Fusion, Tracker Selection, Online Update.

Abstract: This paper presents a new method for combining several independent and heterogeneous tracking algorithms for the task of online single-object tracking. The proposed algorithm runs several trackers in parallel, where each of them relies on a different set of complementary low-level features. Only one tracker is selected at a given frame, and the choice is based on a spatio-temporal coherence criterion and normalised confidence estimates. The key idea is that the individual trackers are kept completely independent, which reduces the risk of drift in situations where for example a tracker with an inaccurate or inappropriate appearance model negatively impacts the performance of the others. Moreover, the proposed approach is able to switch between different tracking methods when the scene conditions or the object appearance rapidly change. We experimentally show with a set of Online Adaboost-based trackers that this formulation of multiple trackers improves the tracking results in comparison to more classical combinations of trackers. And we further improve the overall performance and computational efficiency by introducing a selective update step in the tracking framework.

1 INTRODUCTION

Visual tracking of arbitrary objects in complex scenes is a challenging task that has gained increasing interest in computer vision research during the past years. We consider here the problem of *online* tracking of a single object in a video stream, *i.e*. the object's position at a given time is only estimated from its previous position(s) and previous and current video frames but not from future observations. Also, the camera is not required to be fixed and no prior information about background nor the object's appearance or motion is used. In this setting, the object's shape and appearance and the background can change considerably due to different lighting conditions, viewing angles, deformations, and partial occlusions. To successfully track an object, the algorithm needs to dynamically adapt to the visual changes of the object's appearance as well as its environment. This adaptation however bears the risk of gradually including background information into the object model, which leads the tracker to drift and eventually lose the object.

1.1 Related Work

One way of increasing the robustness to changing appearance and external conditions is by using several different appearance (or motion) models which are (dynamically) selected or merged in a way that makes the tracker less sensitive to these changes at a given time. Some existing methods (Yilmaz et al., 2004; Hua et al., 2006; Stalder et al., 2009) follow this approach by (online) training a foreground/background-classifier based on *several* visual features characterising different visual aspects like colour, texture or shape, *e.g*. Histogram of Oriented Gradients (HOG), colour histograms, filter responses, or local descriptors like SIFT, etc. However, this *low-level* fusion of features leads to problems, when some of the visual attributes are suddenly altered or hidden due to changes in lighting, the object's view point, deformations or partial occlusions, for example. Other methods fuse different modalities in a more explicit way and at a higher level. For example, Collins *et al*. (Collins and Liu, 2005) compute likelihood (or confidence) maps with different linear combinations of RGB channels for each video frame. Then they perform mean shift to track a given object in each likelihood map and fuse the result of each "tracker" using the median. Yin *et al*. (Yin et al., 2008) extend this approach with different features (*e.g*. HOG, saliency) and using adaptive weights depending on the foreground/background separability of each feature. Triesch (Triesch and v. d. Malsburg, 2001) also used dif-

ferent visual cues to produce saliency maps and their reliabilities to integrate them into a tracking result.

Some tracking methods relying on Particle Filters (Perez et al., 2004; Maggio et al., 2007; Badrinarayanan et al., 2007; Moreno-Noguer et al., 2008; Nickel and Stiefelhagen, 2008; Duffner et al., 2009) combine the modalities by using different observation likelihood functions or different state spaces and then integrate these multiple cues in a probabilistic way. Leichter *et al*. (Leichter et al., 2006) proposed a more general probabilistic framework to fuse the output of different independent trackers. More recently, Kwon *et al*. (Kwon and Lee, 2010) proposed in their VTD method to combine different motion and appearance models using Interactive MCMC, and in VTS (Kwon and Lee, 2011) to sample from a space of trackers with different properties in order to increase the overall tracking robustness. However, the interaction between trackers can become relatively complex and difficult to control. Also, many appearance models may need to be evaluated for each video frame, which makes these approaches computationally expensive.

When treating visual tracking as a detection or classification problem, one can make use of so-called *ensemble classifiers* for combining several classifiers to increase the discriminative power (Kittler et al., 1998). Avidan *et al*. (Avidan, 2007), for example, proposed to use Adaboost where several "weak" classifiers are combined into a "strong" one and updated dynamically in order to classify each pixel as background or object. Similarly, Grabner *et al*. (Grabner and Bischof, 2006) use a large dynamic pool of weak classifiers operating on the whole object image patch or parts of it, and only a small part of them is selected at each frame to form the final classifier.

Recently, Bailer *et al*. (Bailer et al., 2014) proposed a method to fuse trackers only by considering their estimated bounding boxes of the object, using a specific energy minimisation framework that takes into account the global performance of each tracker as well as trajectories over time. They show that by combining many existing recent tracking algorithms, the state-of-the-art can be improved.

Finally, the most similar work to ours is the one from Stenger *et al*. (Stenger et al., 2009), where several different tracking methods working with different visual features are combined in a parallel or sequential way based on their normalised confidence values. They test their approach on a set of trackers and all possible pairs and triplets of them. However, the evaluation is performed only on a small set of videos for the tasks of hand and face tracking, and they use an additional hand/face detector, trained off-line. Moreover, in our work, by using the *same* architecture (Online Adaboost) for the *individual* trackers as well as for the baseline methods allows for a more rigorous evaluation of the contribution of individual trackers and the proposed scheme for combining them.

1.2 Motivation

The problem with approaches that use likelihood/confidence maps and also methods based on tracker sampling (*e.g*. VTS) is that the likelihood needs to be computed on many image positions (or at least a search window) and possibly at different scales, which is computationally expensive. Furthermore, existing ensemble classifiers (*e.g*. using Adaboost) often use highly correlated weak classifiers or features, and can only *gradually* adapt to changes in object appearance or scene variations or otherwise suffer from drift. This motivates our choice of combing several independent trackers at a higher level, each of them using features based on different visual aspects like colour, texture and shape. By recurrently selecting the most suitable tracker, the overall system can switch rapidly between appearance models depending on the changes of the scene and the object.

The remainder of this paper is organised as follows. In Section 2, we outline the proposed approach. In Section 3, we briefly explain the individual trackers used in our approach, and in Section 4, the proposed algorithm for tracker selection and update is presented. Experimental results illustrating the superiority of the proposed method compared to standard fusion methods are shown in Section 5.

2 PRINCIPLE APPROACH

The framework of the approach is demonstrated in Fig. 1. The different competing trackers work in parallel, each of them estimating the object's state, *i.e*. a bounding box, and the confidence in this result. In principle, any kind of trackers can be used as long as it provides a confidence measure, score, or probability. Here, we choose trackers relying on the Online Adaboost (OAB) algorithm (Grabner and Bischof, 2006) enhanced with different types of features. We will briefly describe this method and the visual features in the following section. The confidence value is normalised based on parameters that have been trained before on a separate training data set. This normalisation is an important step as the different trackers can be heterogeneous and their confidence values may have different dynamics.

Then, in a given frame, a process of tracker elimination is applied based on the overall temporal and

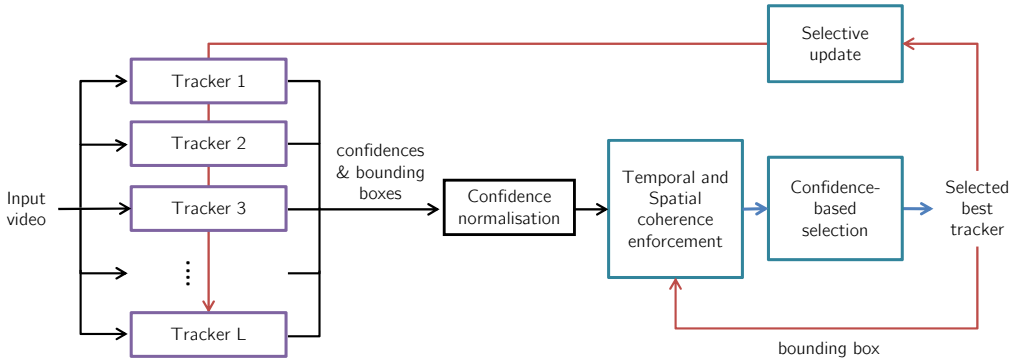

Figure 1: Procedure of the competing tracker approach. One out of L trackers is selected based on their confidence and the spatio-temporal coherence of the solution.

spatial coherence of the result. This filters out the trackers outside a given limit resulting in a smoother and spatially more coherent tracking result. In the final step, the best tracker for the current frame is selected from the remaining individual trackers based on their normalised confidence values. All the trackers keep their individual states (*i.e.* bounding boxes) and are re-considered for selection at each point in time. The advantage of our tracker selection approach is that it avoids unnatural jumps of the resulting bounding boxes while still being able to quickly change from a less confident tracker to more confident one, for example when the object's appearance rapidly changes.

We remind that we do not perform any *fusion* of the individual tracking results which combines the output of all trackers or a subset of them. Also there is no direct interaction between the different trackers. Each tracker remains independent, and only one tracker is selected based on the individual confidence values and the global spatio-temporal coherence.

3 INDIVIDUAL TRACKERS

The concept for this paper is to use relatively simple and fast trackers, that are based on low-level independent and complementary features. Here, we propose to use different OAB trackers, each of them using a different type of visual feature which will be described in Section 3.2. The choice of using Adaboost-based trackers is motivated by their computational efficiency and the simplicity of using different independent features with the same architecture. Of course, a different tracking method (*e.g.* structured output SVMs) can be used as well.

3.1 Online Adaboost

Online Adaboost (OAB) (Grabner and Bischof, 2006) is an extension of the well-known Adaboost algorithm (Freund and Schapire, 1997) for on-line learning that has been effectively applied to the on-line tracking problem (Grabner et al., 2006) by training a binary classifier with foreground and background image patches. In this approach, a "strong" classifier H uses a set of "selectors" $h_n^{sel}, (n \in 1..N)$, each of them iteratively choosing the most discriminative features from a global pool of "weak" classifiers $h_m, (m \in 1..M)$. Each $h_m(\mathbf{x}) \in \{-1,+1\}$ performs a simple binary classification using a single low-level feature extracted from a candidate image region (explained in the following section). Then, the final strong classifier output for a given example image patch $\mathbf{x}$ is:

$$H(\mathbf{x}) = \Phi\left(\sum_{n=1}^{N} \alpha_n h_n^{sel}(\mathbf{x})\right), \qquad (1)$$

where the $\alpha_n \in (0,1)$ are the weak classifier weights:

$$\alpha_n = \begin{cases} 0 & \text{if } e_n < 0.5 \\ \log\left(\dfrac{1-e_n}{e_n}\right) & \text{otherwise,} \end{cases} \qquad (2)$$

with e_n being the classification error of h_n^{sel}. The function $\Phi(\cdot)$ is a decision function (*e.g.* the Heavyside step function), which is of minor importance in our approach as we only use the confidence value (see Section 4.1). At each frame, the object is tracked by applying the classifier several times on image patches in a region around the last tracking position (the search window) and choosing the patch $\mathbf{x}^*$ with the highest confidence (see Section 4.1). Then the feature distributions in the weak classifiers corresponding to object and background as well as the classifiers' weights α_n are incrementally updated based on

the new image information and the respective classification errors. For more details we refer to (Grabner et al., 2006).

3.2 Low Level Features

As mentioned before, each individual tracker relies on different types of features. In this paper, we propose to use three types of complementary features (and three individual trackers) explained in the following. Clearly, the proposed method is not limited to these three, and more trackers and/or other feature types can be added. A feature is computed on a subregion of the image patch that corresponds to the object's bounding box.

- **Haar-like Features:** as originally proposed in (Grabner et al., 2006; Viola and Jones, 2001). They correspond to differences of the mean pixel intensities of adjacent rectangular regions.
- **Histogram of Oriented Gradient (HOG) Features:** a simplification of the original HOG (Dalal and Triggs, 2005) where the gradients computed on a region are quantised into a set of 8 orientations and 3 magnitudes, and the value of a single bin is used as a feature.
- **Histogram Of Colour (HOC) Features:** HSV histograms of a region are computed with a $2 \times 3 \times 8$ quantisation on the Hue and Saturation channels. As for the HOG features, the value of a single bin is used as a feature.

The feature selection (sub-regions, bins) is performed by the standard Adaboost algorithm on a subset of random samples.

4 TRACKER COMBINATION

Each of the individual trackers alone does not perform very well on average, measured on a challenging tracking benchmark data set. However, as we will show experimentally, our proposed combination scheme of multiple trackers using different types of features outperforms each of these individual trackers as well as a *single* tracker that combines *all* types of features in the classical way.

4.1 Confidence Measure and Normalisation

A variety of confidence measures exist depending on the tracking or classification algorithm. Here, we use the one originally proposed for OAB. Given a set of trackers $T_k, (k \in 1..L)$, the confidence c_k for T_k is obtained by weighting the results of each one of its selectors h_n^{sel}. For an example $\mathbf{x}$, we have:

$$c_k(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n h_n^{sel}(\mathbf{x}) . \quad (3)$$

This confidence measure expresses, in a way, the proportion of selectors h_n^{sel} having correctly classified example $\mathbf{x}$. Note that it is directly related to the output of the strong classifier (Eq. 1).

Having different trackers, the confidence values for each one will have different dynamics. In order to be able to compare the confidence values, we normalise them for each tracker T_k using the mean μ_k and standard deviation σ_k computed on a separate data set: ALOV300++ (Smeulder et al., 2014), similar to (Stenger et al., 2009): $c'_k = \frac{c_k - \mu_k}{\sigma_k}$. These videos cover various changes in illuminations, transparency, clutter, occlusion, zoom, appearance, motion patterns and contrast, which makes them suitable for estimating μ_k and σ_k in typical real-world scenarios.

4.2 Spatial and Temporal Coherence

Using the normalised confidence to choose the best tracker at a given frame t is not sufficient, as we will see in the experiments section. When the bounding boxes of the individual trackers are further apart (because one or several of them have drifted away), jumps can occur in the overall tracking process since no continuity is present in the choice of best confidence. In order to avoid these jumps and to make the output smoother, a spatial and temporal coherence criterion is introduced, which ignores the result of those trackers that are too far away from the previous object's position.

At each frame t, we have a set of trackers $T_k, (k \in 1..L)$, that given an example, return a confidence c_k and a bounding box B_t^k surrounding the object in the image. This bounding box is defined by the position of its centre (x_t^k, y_t^k) and its dimensions (w_t^k, h_t^k). The bounding box B_{t-1} of the previous frame $t-1$ coming from the *selected* tracker is saved and used to compute its distance to each of the current tracker's bounding boxes B_t^k. Then, at frame t, tracker T_k is (temporarily) eliminated if:

$$max\left(\left|x_t^k - x_{t-1}\right| - \Theta_x, \left|y_t^k - y_{t-1}\right| - \Theta_y\right) > 0 . \quad (4)$$

$\Theta = (\Theta_x, \Theta_y)$ is a two-dimensional distance threshold proportional to the size of B_{t-1}, that is $\Theta_x = \beta \frac{w_{t-1}}{2}$, and $\Theta_y = \beta \frac{h_{t-1}}{2}$. The optimal coefficient β is computed empirically by running the algorithm on the ALOV300++ data set and left constant for all of our experiments.

After this step, only a subset of trackers $T_i, (i \in 1..L')$, with $L' \leq L$ is left. The final decision on the best tracker T_s with bounding box B_t is given by selecting the tracker with maximal normalised confidence c'_s:

$$s = \underset{i \in 1..L'}{\operatorname{argmax}} \left(c'_i\right) \quad (5)$$

In the case where $T_i, (i \in 1..L')$ is empty, then only the maximal confidence criterion is applied.

4.3 Selective Update

In the original OAB architecture, the trackers are completely independent in the sense that each tracker $T_k, (k \in 1..L)$, updates its discriminative model using its resulting bounding boxes B_t^k at frame t.

One update scheme would be to use the *selected* tracker's bounding box to update the remaining trackers. This idea proved to be ineffective due to the rapid drift of the now co-dependent trackers. In fact, the trackers start drifting when the background information is falsely included into the foreground model.

Instead we introduce a selective update strategy with a slower update rate (and still independent trackers) giving us a slower learning rate of the models and reduced computational time, which will help prevent this drifting phenomenon. To this end, we choose to only update the previously *selected* tracker with its resulting bounding box, since we consider that the non-selected trackers may have *eventually* drifted. Thus we do $L-1$ less updates than in the original learning architecture.

5 EXPERIMENTS

The experiments for the proposed approach are conducted on the VOT2013 data set (Kristan et al., 2013). VOT2013 is a visual object tracking challenge held in 2013 in order to benchmark on-line tracking algorithms. The data set contains 23 videos and 8416 frames. We did not use the provided VOT2013 evaluation framework because the capability of our proposed fusion method to return to the object after losing it is not taken into account in the VOT2013 framework but on contrary is penalised as the trackers are stopped and reinitialised from the point of loss. Further, the aim here is not necessarily to show that the proposed method outperforms existing state-of-the-art tracking algorithms. We rather want to evaluate and show the benefit of our tracker selection approach compared to classical tracker fusion methods.

5.1 Compared Methods

To be able to correctly evaluate the performance of the proposed approach and to comprehend the impact of the different components, we compared it to a certain number of baseline methods. In the following, we describe the different methods that have been tested.

- **Proposed Method (PM):** As described above, the proposed approach uses $L = 3$ trackers denoted HAAR, HOG and HOC with 50 selectors each. The spatio-temporal coherence and normalised confidence is used for tracker selection. All the trackers are updated every frame.
- **Proposed Method with Selective Update (PM+).** This method is the same as PM, but with the selective update strategy described in Section 4.3.
- **Best Confidence (BC).** Using the same $L = 3$ trackers as the proposed method, the selection of the best tracker is only based on the normalised confidence values.
- **Fusion of Features (FoF).** The OAB method allows different sets of low-level features to be used together in the global pool of weak classifiers. In the FoF baseline, one strong classifier is constructed based on all the different feature types. This type of feature fusion is a classical way of combining different types of low-level observations. In order to provide for a fair comparison, we used the same number of features as in the proposed approach, *i.e.* 3×50 selectors.
- **Fusion of Features with Minimal Update (FoF+).** To be able to correctly compare to the proposed method with selective update, we decreased the number of updates of FoF by $L-1$. To this end, the tracker is only updated one frame out of L.
- **Centroid of Trackers (CoT).** This is a high-level fusion approach, where the resulting bounding box is computed from the mean of the bounding boxes of the individual trackers weighted by their respective confidence values.

5.2 Evaluation Measures

To evaluate the methods mentioned above, we used the common F-score which combines the measures of precision and recall. Based on the areas of the resulting bounding box B_t, the ground-truth box B_t^{gr} and their intersection, the F-score f_t at frame t is defined as:

$$f_t = 2 \times \frac{precision_t \times recall_t}{precision_t + recall_t} = 2 \times \frac{\bigcap(B_t, B_t^{gr})}{B_t + B_t^{gr}} \quad (6)$$

Table 1: Success rates for the different methods on the VOT2013 data set.

method	success rate
Best theoretical selection	*(93.43%)*
Best Confidence (BC)	78.32%
Fusion of Features (FoF)	80.13%
Centroid of Trackers (CoT)	61.21%
Proposed Method (PM)	**81.10%**
Best theor. sel. + selective update	*(93.86%)*
Fusion of Features + min. update (FoF+)	77.45%
Proposed Method + selec. update (PM+)	**83.93%**

Table 2: Prediction rates for Best Confidence and the proposed method on the VOT2013 data set.

method	prediction rate
Best Confidence (BC)	76.22%
Proposed Method (PM)	**77.99%**

with $precision_t = \frac{\bigcap(B_t, B_t^{gr})}{B_t}$ and $recall_t = \frac{\bigcap(B_t, B_t^{gr})}{B_t^{gr}}$.

For each video, the trackers are initialised with the ground truth in the first frame and run until the end of the video. At each frame, the different methods return the F-score associated with their result. The tracking method is considered lost when $f_t = 0$.

Using the F-score, two performance measures are introduced: *success rate* and *prediction rate*. The success rate represents the number of frames having $f_t > 0.1$. The relatively low threshold captures the overall tracking robustness of the method, not so much its bounding box accuracy. The prediction rate on the other hand represents the proportion of frames where the method correctly predicted the best tracker. In fact, knowing the ground truth position of the object, the F-score of each individual tracker is computed, then the best theoretical tracker (and F-score) at each frame in known. Based on this, we can compute the number of frames where a selection method has correctly predicted the best individual tracker. We allow for a 20% margin on the best F-score, *i.e*. the two best trackers are considered equivalent if the difference of their F-score is lower than 20%.

To provide an upper bound, we also computed the success rate of a best theoretical selection, *i.e*. the success rate of a method that always predicts the tracker with the highest F-score. This measure is important as it expresses the general performance of the individual trackers. All presented numbers are averaged measures over 10 tracker runs.

5.3 Results

As shown in Table 1 and Table 2, the success and prediction rate of our proposed method (PM) outperforms Best Confidence (BC). In Fig. 2 and Fig. 3, we also introduce some results from the video 'bolt'. It is a particularly challenging sequence due to the number of similar objects and colours, complex background, and change in appearance. We can see in Fig. 3 (1) that the use of our spatio-temporal coherence criterion eliminates the jumps that are introduced when only using the normalised confidence BC. Although surpassing BC, PM cannot avoid all the jumps. However, thanks to the simultaneous use of the spatio-temporal coherence and the confidence, the proposed method has the ability to relock on the object without any reinitialisation of the trackers. This can be illustrated in Fig. 2 (a), where we can observe that the tracker HOC (green) is the only one able to correctly track the object. BC (b) has a very discontinuous tracking while PM (d) succeeded to lock on the HOC tracker.

The proposed method also outperforms the Fusion of Features (FoF) approach. In this approach, the features are not independent, since they are all updated with the same data. As a consequence, as soon as a tracker begins to drift, wrong data or noise is introduced, and it is impossible for the trackers to recover as illustrated in Fig. 4 and Fig. 2 (c) : The FoF method started drifting in frame 5, and completely loses the object afterwards. Unlike this approach, PM (d) uses independent trackers. Thus, the selection scheme makes it possible to switch from one tracker to another if it is lost. PM succeeds in tracking the object at almost the performance of the best theoretical results (Fig. 3).

As for the Centroid of Trackers, the low success rate (Table 1) shows that it is not a valuable method as the individual trackers do not always perform well simultaneously, and a lost tracker may considerably disturb the overall result.

Finally, the bottom of Table 1 shows the success rates for the proposed method with selective update (PM+) and the corresponding baseline FoF+. The selective update slightly increases the performance and produces the best results among the compared methods. In Fig. 3 (2), we can see that PM+ successfully tracks the object whereas the FoF+ only tracks correctly for a certain time and then starts drifting and loses the object. The selective update also improves the speed by around 50%. The proposed method runs at around 6 fps (with non-optimised and single-core C++ code), where the majority of computation time is spent on feature computation and update of the Adaboost trackers. The combination method using temporal and spatial coherence with confidence (PM) adds very little computational overhead.

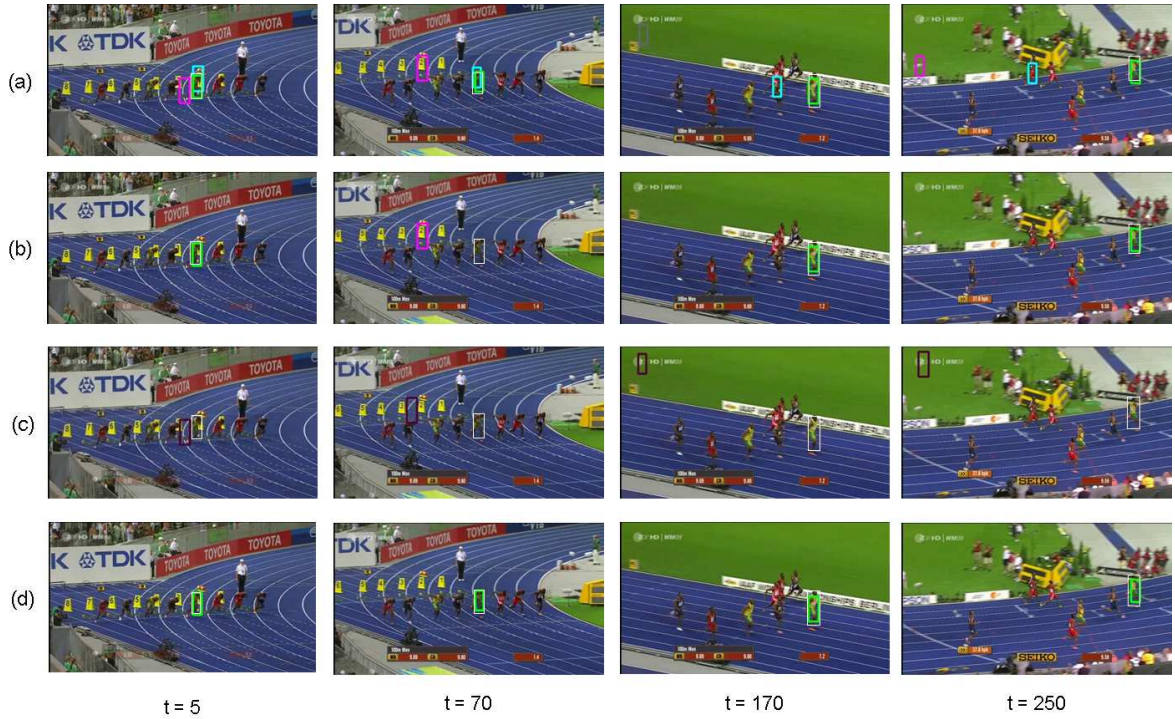

Figure 2: Comparison of tracking results on the "bolt" video for (a) the individual trackers (*white*: ground-truth, *pink*: Haar tracker, *blue*: HOG tracker, *green*: HOC Tracker), (b) Best Confidence (BC), (c) Fusion of Features (FoF), and (d) the proposed method (PM).

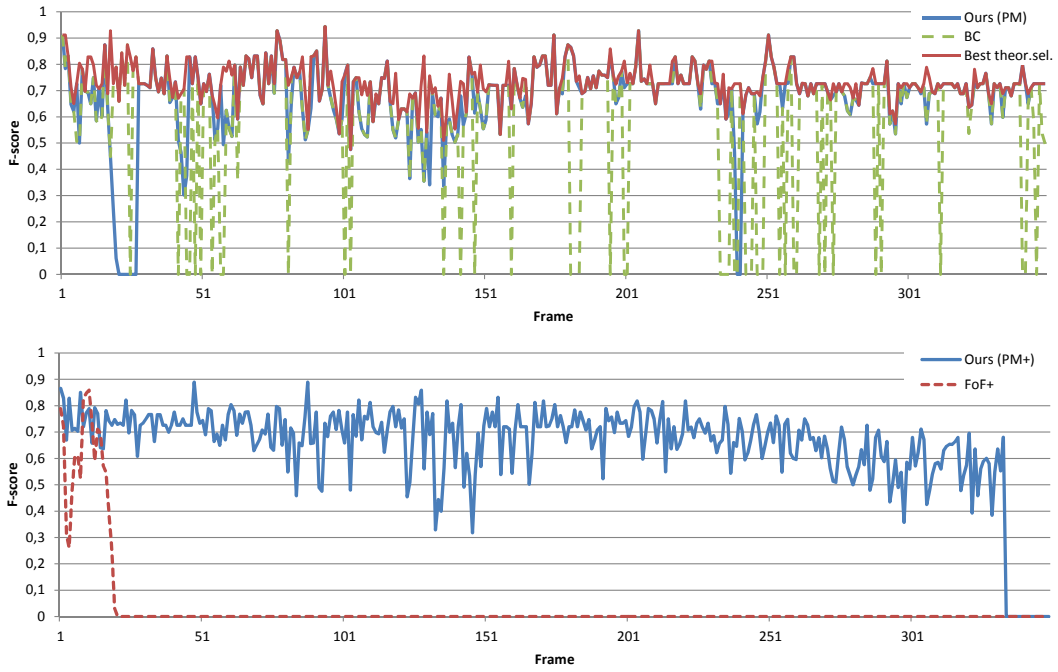

Figure 3: F-score measures of the "bolt" video, (1) proposed method (PM), Best confidence (BC), best theoretical results and (2) proposed method with selective update (PM+), Fusion of Features with minimal update (FoF+).

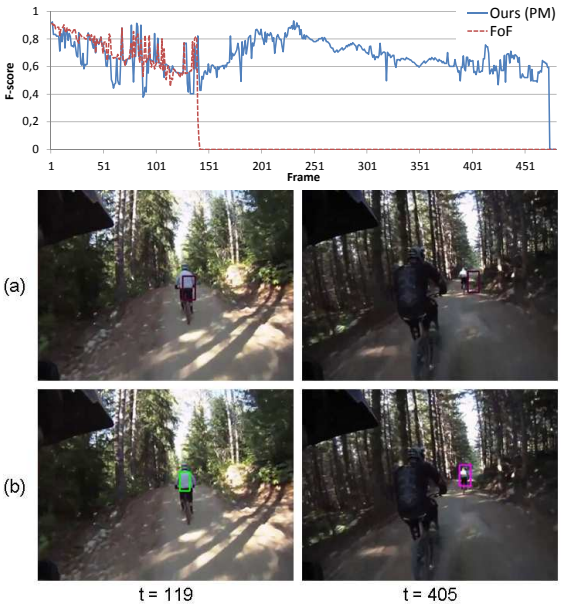

Figure 4: Comparison of results on the "dh" video for (a) Fusion of Features (FoF), (b) the proposed method (PM) where (*pink*: Haar, *blue*: HOG, *green*: HOC Tracker).

6 CONCLUSION

In this paper, we presented a simple method to select from a pool of trackers the most suitable one. It integrates a spatial and temporal coherence criterion, a consistent confidence evaluation for tracker selection and a selective update strategy. We used OAB-based trackers with simple low-level features. Experimental results demonstrate that, even in very challenging sequences, the proposed method improves the overall robustness and outperforms classical tracker combination strategies.

Future work will concentrate on introducing new and better performing trackers also introducing more low-level features (*e.g.* motion) to complete the set of trackers and achieve a best theoretical selection rate of 100% (*c.f.* Table 1).

REFERENCES

Avidan, S. (2007). Ensemble tracking. *IEEE Trans. on PAMI*, 29(2):261–271.

Badrinarayanan, V., Perez, P., Le Clerc, F., and Oisel, L. (2007). Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues. In *Proc. of ICCV*.

Bailer, C., Pagani, A., and Stricker, D. (2014). A superior tracking approach: Building a strong tracker through fusion. In *Proc. of ECCV*, pages 170–185.

Collins, R. T. and Liu, Y. (2005). On-line selection of discriminative tracking features. *IEEE Trans. on PAMI*, 27(10):1631–1643.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. of CVPR*.

Duffner, S., Odobez, J.-M., and Ricci, E. (2009). Dynamic partitioned sampling for tracking with discriminative features. In *Proc. of BMVC*, London, UK.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Grabner, H. and Bischof, H. (2006). On-line boosting and vision. In *Proc. of CVPR*, pages 260–267.

Grabner, H., Grabner, M., and Bischof, H. (2006). Real-time tracking via on-line boosting. In *Proc. of BMVC*, pages 47–56.

Hua, C., Wu, H., Chen, Q., Wada, T., and City, W. (2006). A pixel-wise object tracking algorithm with target and background sample. In *Proc. of ICPR*.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). On combining classifiers. *IEEE Trans. on PAMI*, 20(3):226–239.

Kristan, M., Cehovin, L., Pflugfelder, R., Nebehay, G., Fernandez, G., Matas, J., and et al. (2013). The Visual Object Tracking VOT2013 challenge results. In *Proc. of ICCV (Workshops)*.

Kwon, J. and Lee, K. (2010). Visual tracking decomposition. In *Proc. of CVPR*, pages 1269–1276.

Kwon, J. and Lee, K. (2011). Tracking by sampling trackers. In *Proc. of ICCV*.

Leichter, I., Lindenbaum, M., and Rivlin, E. (2006). A general framework for combining visual trackers – "black boxes" approach. *IJCV*, 67(3):343–363.

Maggio, E., Smeraldi, F., and Cavallaro, A. (2007). Adaptive multifeature tracking in a particle filtering framework. *IEEE on Circuits and Systems for Video Technology*, 17(10):1348–1359.

Moreno-Noguer, F., Sanfeliu, A., and Dimitris, S. (2008). Dependent multiple cue integration for robust tracking. *IEEE Trans. on PAMI*, 30(4):670–685.

Nickel, K. and Stiefelhagen, R. (2008). Dynamic integration of generalized cues for person tracking. In *Proc. of ECCV*, pages 514–526.

Perez, P., Vermaak, J., and Blake, A. (2004). Data fusion for visual tracking with particles. *Proc. of IEEE*, 93(3):495–513.

Smeulder, W. M. A., Dung, M. C., Cucchiara, R., Calderara, S., Deghghan, A., and Shah, M. (2014). Visual tracking: an experimental survey. *IEEE Trans. on PAMI*.

Stalder, S., Grabner, H., and Gool, L. V. (2009). Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *ICCV (WS on On-line Comp. Vis.))*, pages 1409–1416.

Stenger, B., Woodley, T., and Cipolla, R. (2009). Learning to track with multiple observers. In *Proc. of CVPR*.

Triesch, J. and v. d. Malsburg, C. (2001). Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. of CVPR*, volume 1.

Yilmaz, A., Li, X., and Shah, M. (2004). Object contour tracking using level sets. In *Proc. of ACCV*.

Yin, Z., Porikli, F., and Collins, R. T. (2008). Likelihood map fusion for visual object tracking. In *IEEE Workshop on Applications of Computer Vision*, pages 1–7.