Classifying Global Scene Context for On-line Multiple Tracker Selection

Salma Moujtahid
salma.moujtahid@liris.cnrs.fr
Stefan Duffner
stefan.duffner@liris.cnrs.fr
Atilla Baskurt
atilla.baskurt@liris.cnrs.fr

Université de Lyon, CNRS, INSA-Lyon,
LIRIS, UMR5205,
F-69621, France

Abstract

In this paper, we present a novel framework for combining several independent on-line trackers using visual scene context. The aim of our method is to decide automatically at each point in time which specific tracking algorithm works best under the given scene or acquisition conditions. To this end, we define a set of generic global context features computed on each frame of a set of training videos. At the same time, we record the performance of each individual tracker on these videos in terms of object bounding box overlap with the ground truth. Then a classifier is trained to estimate which tracker gives the best result given the global scene context in a particular frame. We experimentally show that such a classifier can predict the best tracker with a precision of over 80% in unknown videos with unknown environments. The proposed tracking method further filters the classifier responses temporarily using a Hidden Markov Model in order to avoid rapid oscillations between different trackers. Finally, we evaluated the overall tracking system and showed that this scene context-based tracker selection considerably improves the overall robustness and compares favourably with the state-of-the-art.

1 Introduction

We consider the problem of on-line visual object tracking in unconstrained environments, which raises many challenges. First, the arbitrary nature of the object makes it difficult to model its appearance only from the first video frame and implies the use of on-line learning as opposed to off-line trained classifiers for specific types of objects (*e.g.* faces or pedestrians). Additionally, the potential changes in the scene appearance, camera motion or other acquisition conditions during a video make background subtraction algorithms unsuitable for the tracking task. In order to successfully track an object in these unconstrained conditions, a tracking algorithm needs to adapt its model based on discriminant, descriptive features while minimising model drift. In our work, we leverage the fact that different tracking algorithms are *specialised* on different types of scenes and acquisition conditions. We show that, by quantifying these scene context parameters and training a discriminative classifier that decides at each moment which specific algorithm will perform best under the given conditions, we are able to successfully combine several independent trackers in a way that considerably improves the overall robustness in highly varying environments.

Pages 163.1-163.12
DOI: https://dx.doi.org/10.5244/C.29.163

1.1 Related Work

In the literature, many ways of combining, fusing or selecting visual features have been presented. An example of low-level fusion of features is the Bayesian framework introduced by Yilmaz *et al.* [29] fusing probabilistic density functions based on texture and colour features for object contour tracking. Collins *et al.* [4] used likelihood maps to rank features in order to select the most discriminant one. Other existing works (*e.g.* [20, 23, 26, 30]) fuse different modalities, like motion or shape, in order to improve the overall foreground-background discrimination. Fusion is also possible at a higher level, where several trackers are run in parallel in order to select or combine their respective results. A probabilistic combination has been proposed by Leichter *et al.* [16] where parallel trackers use different features and output a probabilistic density function of the tracked state. Another probabilistic formulation has been presented by Kwon *et al.* [14, 15], where different motion and appearance models are combined by sampling from different trackers. More recently, the method Bailer *et al.* [2] fuses the bounding boxes of multiple state-of-the-art trackers by an off-line training step and trajectory optimisation. In terms of model or feature fusion, our previous work Moujtahid *et al.* [18] concentrated on using confidence values of several individual trackers to select the most suitable one at a given instant. Each tracker is independent and relies on a different visual feature, like colour or texture. The similar approach from Stenger *et al.* [24] also used confidences but has been applied to the particular case of face tracking and involves off-line trained classifiers, whereas in [18] we used an additional spatial-temporal coherence criteria to enforce the continuity of tracking.

In contrast to these existing works, in our proposed framework, the selection of the most discriminant and most suitable tracker is based on the *visual scene context* in the video. Context has been used previously for object tracking in different forms and has shown to improve the overall tracking performance. Some works propose to detect image regions or interest points that move similarly to the tracked object [9, 27, 28] in order to assist the tracking, so-called supporters, contributors or helper objects. Other methods seek image regions that have similar appearance, so-called distractors, in order to avoid confusion [6, 11]. However, these approaches are computationally expensive, due to the more complex data association and modelling of spatial and temporal relationships between the different tracked objects or interest points.

In our approach, we are not trying to detect and track supporting or distracting image regions but we are classifying the general scene context and conditions in order to select the most appropriate visual cue or tracker for a given situation. To this end, we compute global image descriptors based on colour, intensity and motion at each video frame. In the past, other global image descriptors (sometimes called gist features) have been proposed (*e.g.* [19, 21, 25]) mostly for fixed images to classify scenes into different semantic categories, such as open, closed environments, indoor, outdoor *etc*. To our knowledge, no other work exists that extracts and classifies global scene context features for visual object tracking.

1.2 Motivation

Different tracking algorithms have different strengths and weaknesses. Some cope well with different lighting conditions, some are particularly robust to object deformations or occlusions *etc*. but will fail in other conditions. Due to the compromise between invariance and discriminative power, despite recent progress in on-line tracking algorithms, it is hard to design models that perform well in very different environments and contexts. Although the

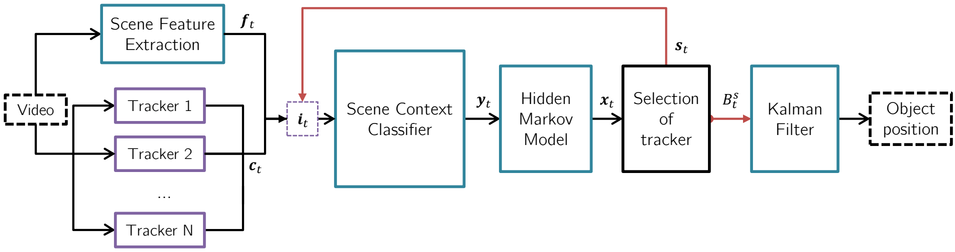

Figure 1: Overall framework of the proposed scene context-based tracking algorithm.

fusion of different visual cues generally improves the performance in that regard, it remains difficult to decide on the importance or on the weight each modality should get when the overall scene context changes, especially *within* a given video. Moreover, the on-line updating of the underlying models is not straightforward. A given modality or tracker might be impaired by the update under conditions that it has not been designed for.

To this end, we propose a framework that combines several independent and complementary trackers, each specialised on different image and scene conditions. The decision on which tracker to select is proposed by an off-line trained classifier which, in turn, is based on general scene context features that are independent from the individual trackers. To summarise, our contributions are the following:

- a set of *general scene context features* that describe the global conditions (such as lighting, camera motion) that are relevant to track an object in a video,
- a framework that combines independent trackers by using a *classifier* that estimates at each frame the most suitable tracker for the given context, and by filtering the classifier responses using a temporal Hidden Markov Model (HMM),
- and a *thorough evaluation* of the different components of the proposed approach and comparison with state-of-the-art methods.

2 Overall approach

The general procedure of the proposed tracking framework is illustrated in Fig. 1. On a given video, N independent trackers $T_n, (n \in 1..N)$ run in parallel and, at every frame t, produce each an estimate of the object's state. This is usually a bounding box B_t^n with an associated confidence value (or score) $c_{t,n}$. The objective is to select at each frame the best tracker, *i.e.* the one that outputs the bounding box that fits best the object to track.

At the same time, M scene context features $\mathbf{f}_t$ are extracted and concatenated with additional measures like the trackers' confidences $\mathbf{c}_t$ and the identifier of the last selected tracker s_{t-1} to form a large feature vector $\mathbf{i}_t$. An N-class classifier, that has been trained off-line on annotated data, is then applied on these features to estimate the best tracker for the given scene context. The classifier responds with $\mathbf{y}_t$, a probability for each class which is subsequently filtered by a HMM to ensure some temporal continuity of the tracker selection and reject outliers. Finally, a Kalman Filter is applied as a post-processing step to temporally smooth the resulting object bounding box B_t^s from the selected trackers T_s. The result of the Kalman filter represents the final output of our tracking algorithm, and is further used to update the models of the individual trackers T_n.

Apart from this last update step, all the trackers are completely independent and do not cooperate or interact with each other. It is also important to mention that this approach is very generic, and in theory *any* on-line tracking algorithm can be integrated in this framework.

3 Visual scene context

In most existing computer vision algorithms, for example image classification or object recognition, some common visual features are used to concisely describe the content of an image or a region, such as SIFT, HOG, Haar-like features, LBP, or more roughly using Gist descriptors. However, to our knowledge, there are no established descriptors that capture global scene information regarding the overall environment or acquisition conditions, like lighting, camera motion, background uniformity *etc*. To quantify these phenomena is of particular interest for video analysis and object tracking algorithms. In the following, we will propose a set of such descriptors that we call “scene context features”.

3.1 Scene context feature extraction

The proposed scene context features are designed to help predicting the best tracker in a given environment. Most of them correspond to first and second order statistics of a given image-related variable (*e.g*. intensity, hue, saturation, motion vectors), and are straightforward to compute. Let’s define Ω as a region of the input image, and $f^{\Omega}_{t,k}$ as our feature computed on Ω. To simplify the notation, we omit the frame index t in this section. We propose to use the following set of features, grouped into three categories and defined in Equations 1-4:

Intensity features

- *Average brightness* (f^{Ω}_1): the mean grey-scale pixel value over region Ω (see Eq. 1).
- *Average contrast* (f^{Ω}_2): the mean squared value of the difference of each grey-scale pixel and the average brightness over Ω (see Eq. 4).

Chromatic features

- *Average saturation* (f^{Ω}_3): the mean pixel value of the saturation channel in HSV colour space over the region Ω (Eq. 1).
- *Saturation variance* (f^{Ω}_4): the variance of saturation over Ω (Eq. 2).
- *Dominant hue* (f^{Ω}_5): the dominant colour of the region Ω extracted from a histogram of quantised hue pixel values in HSV colour space (Eq. 3).
- *Hue variance* (f^{Ω}_6): the variance of the pixel values in the hue channel (Eq. 2).

Motion features

- *Average motion* (f^{Ω}_7): the mean of the norm of optical flow vectors densely computed over the region Ω (Eq. 1).
- *motion variance* (f^{Ω}_8): the variance of the norm of dense optical flow vectors over Ω (Eq. 2).

Let p_i denote the pixel value of a given image channel (*e.g.* H,S,V) and $||\Omega||$ the number of pixels in the region Ω. The above mentioned features are then defined as follows:

$$\text{AVERAGE:} \quad f_k^\Omega = \frac{\sum_{i\in\Omega} p_i}{||\Omega||}, \qquad \text{for } k = 1,3,7 \tag{1}$$

$$\text{VARIANCE:} \quad f_k^\Omega = \frac{\sum_{i\in\Omega} (p_i)^2}{||\Omega||} - \left(\frac{\sum_{i\in\Omega} p_i}{||\Omega||}\right)^2, \qquad \text{for } k = 4,6,8 \tag{2}$$

$$\text{DOMINANT CUE:} \quad f_k^\Omega = \underset{i\in\Omega}{\operatorname{argmax}}(p_i), \qquad \text{for } k = 5 \tag{3}$$

$$\text{CONTRAST:} \quad f_k^\Omega = \frac{1}{||\Omega||} \cdot \sum_{i\in\Omega} \left(p_i - \frac{\sum_{i\in\Omega} p_i}{||\Omega||}\right)^2, \qquad \text{for } k = 2. \tag{4}$$

Each of these features k is computed on three different image regions Ω. We define a **global value** as the feature computed on the whole image: f_k^G. The **local value** is the feature computed on the Region Of Interest (ROI), *i.e.* the region defined by the bounding box of the tracked object: f_k^L. And a **differential value** as the difference between the feature computed on the foreground region (*i.e.* the ROI) and the background region (*i.e.* the image not including the ROI): f_k^D.

Not every combination of feature and region is used as some them don't show semantic meaning. The concatenation of these features gives us:

$$\mathbf{f}^G = \{f_1^G, \dots, f_3^G, f_6^G, \dots, f_8^G\} \qquad \mathbf{f}^L = \{f_1^L, \dots, f_8^L\} \qquad \mathbf{f}^D = \{f_1^D, \dots, f_7^D\}.$$

Finally, we obtain $M = 21$ scene context features $\mathbf{f}_t = \{\mathbf{f}_t^G, \mathbf{f}_t^L, \mathbf{f}_t^D\}$ for frame t.

3.2 Scene Context Classifier

To learn the different patterns that show high correlation between the information extracted from the scene context and the performance of a tracker in a particular set of conditions, we employ a multi-class classifier. We chose a fully connected Multi-Layer Perceptron (MLP) with one hidden layer of N_h neurons and N output neurons. Any other algorithm could be used. In fact, a multi-class SVM showed equivalent performance, in our experiments. However, it was relatively sensitive to the choice of hyper-parameters (*e.g.* the type of kernel).

The input to the classifier at frame t consists of the scene features $\mathbf{f}_t$ and two additional components: First, the *confidence* values of the N trackers $\mathbf{c}_t = (c_{t,1}..c_{t,N})$; they provide the classifier with a measure of reliability of each tracker's result. And second, the *identifier* s_{t-1} of the tracker that has been selected in the previous frame. We will experimentally show that this recursion highly contributes to learning the correlation between the scene context features and the selected tracker in a given frame.

Furthermore, in order to give the scene context classifier information on the evolution of the context over time, we additionally provide it with the features from the two previous frames $t-1$ and $t-2$ forming the vectors:

$$\begin{aligned} \mathbf{F}_t &= \{\mathbf{f}_t, \mathbf{f}_{t-1}, \mathbf{f}_{t-2}\} \\ \mathbf{C}_t &= \{\mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{c}_{t-2}\} \\ \mathbf{S}_{t-1} &= \{s_{t-1}, s_{t-2}, s_{t-3}\} \end{aligned}$$

Incorporating the temporal aspect has proven to be very effective, as shown in our experimental results. The final feature vector given as input to the classifier is the following:

$$\mathbf{i}_t = \{\mathbf{F}_t, \mathbf{C}_t, \mathbf{S}_{t-1}\}.$$

The classifier is trained off-line with a set of training samples $\left\{\mathbf{i}_j, o_j^*\right\}_{j=1}^{N_t}$ containing the input features $\mathbf{i}_j$ and labels o_j^* computed from a separate set of videos with annotated object bounding boxes. To construct the scene context features $\mathbf{i}_j = \left\{\mathbf{F}_j, \mathbf{C}_j, \mathbf{S}_{j-1}\right\}$, we run the N trackers on each video, and at each frame, extract the scene context features $\mathbf{F}_j$ as well as the trackers' confidences $\mathbf{C}_j$. For the vector $\mathbf{S}_{j-1}$, *i.e.* the previously selected tracker, we used the identifiers of the best tracker in the respective preceding frames according to the ground truth. The desired classifier output class o_t^* is the most accurate tracker, *i.e.* the one with the highest overlap with the ground truth. We optimise the neural network parameters with standard stochastic gradient descent by minimising the mean squared error between the networks response vector $\mathbf{y}_t = \{y_{t,1}..y_{t,n}\}$ and the desired output vector $\mathbf{y}_t^* \in \{-1,+1\}^N$ with $+1$ for the component corresponding to o_t^* and -1 otherwise. The network's final class prediction is simply $o_t = \mathrm{argmax}_{n \in N} \mathbf{y}_{t,n}$. We perform early stopping using a separate validation set. At each validation step, we update the component s (the previously selected tracker) in the input vectors according to the maximum of the actual output of the classifier.

4 Tracking procedure

The proposed algorithm uses N independent on-line trackers that are initialised with the bounding box of the object in the first video frame. Then, as illustrated in Fig. 1, the trackers and the context feature extraction operate in parallel providing at each frame N confidence values $\mathbf{C}_t$ and M scene context features $\mathbf{F}_t$ respectively.

At each video frame, the scene context classifier estimates the best tracker $\mathbf{y}_t$ to select for the given scene context. To avoid frequent and unnecessary switching between different trackers, we filter the classifier responses in time using a Hidden Markov Model (HMM).

The HMM is used to estimate the discrete hidden variable $x_t \in \{1..N\}$ corresponding to the best tracker selection, and it receives the observations $\mathbf{y}_t$ being the output of the scene context classifier. Another observation variable $\mathbf{d}_t = (d_{t,1}..d_{t,N})$ is added and defined as the normalised distances of each tracker's resulting bounding box B_t^n to the previous estimated object position. Using the HMM, we want to estimate the posterior probability distribution of x_t, which we can compute recursively:

$$p(x_t|\mathbf{y}_t,\mathbf{d}_t) = p(\mathbf{y}_t|x_t)\int p(x_t|x_{t-1},\mathbf{d}_t)p(x_{t-1}|\mathbf{y}_{t-1},\mathbf{d}_{t-1})\mathrm{d}x_{t-1}\,. \tag{5}$$

To simplify model parameter estimation, we assume that observations $\mathbf{d}_t$ and the hidden variable x_{t-1} are independent. This gives us:

$$p(x_t|\mathbf{y}_t,\mathbf{d}_t) = p(\mathbf{y}_t|x_t)p(x_t|\mathbf{d}_t)\int p(x_t|x_{t-1})p(x_{t-1}|\mathbf{y}_{t-1},\mathbf{d}_{t-1})\mathrm{d}x_{t-1} \tag{6}$$

The likelihood function $p(\mathbf{y}_t|x_t)$ of the observed classifier responses and the probability $p(x_t|\mathbf{d}_t)$ of selecting a tracker given the distances to the previous bounding box are modelled by histograms computed on a separate training set. The transition probability $p(x_t|x_{t-1})$ is set empirically to achieve a reasonable continuity of the HMM responses. Then, the best tracker selection according to the HMM is computed as the maximum a posteriori probability:

$$s_t = \underset{s'}{\mathrm{argmax}}\, p(x_t = s'|\mathbf{y}_t,\mathbf{d}_t)\,. \tag{7}$$

and the bounding box B_t from the selecting tracker T_{s_t} is passed to a Kalman Filter to provide a smoother final trajectory.

5 Experiments

For our experiments, we used $N = 3$ On-line AdaBoost (OAB) trackers [8] with different visual cues for each: Haar-like features (HAAR), Histograms of Oriented Gradients (HOG) and Histograms of Colour (HOC). The evaluation of the performance of our framework consists of two parts. First, we measured the classification rate of the proposed scene context classifier when trained on the different groups of features described in Section 3.1. Secondly, we evaluated the overall tracking algorithm on a public benchmark analysing the contribution of the different components, *i.e.* scene context features, HMM, and Kalman filter.

Training dataset. The context-based classifier is trained on the Princeton Tracking Benchmark Dataset [22]. It contains 100 RGB-D videos with a diverse set of object types, backgrounds, and changes in illumination, appearance and motion.

Evaluation dataset. The evaluation of context-based classifier, as well as for the overall tracking framework was conducted on the publicly available Visual Object Tracking (VOT2013) benchmark [13]. The VOT2013 dataset contains 16 image sequences collected from well-known tracking evaluations, they cover most of the challenging situations in object tracking: scale variance, complex backgrounds, occlusions, object deformation *etc*.

5.1 Classifier evaluation

In order to understand the relevance of the different scene context features, we conducted a series of experiments related to the classifier. We first separated the scene features into the three group presented in Section 3.1: *local*, *global* and *differential*. Then we added several time steps $\mathbf{F}_t$, the confidences $\mathbf{C}_t$ and the previous tracker identifiers $\mathbf{S}_{t-1}$. For each feature set, the classifier is trained on the Princeton dataset and tested on the VOT2013 dataset. The results for the different combinations are shown in Table 1. The recognition rate represents the proportion of frames where the classifier has successfully predicted the best tracker.

The combination of *local*, *global* and *differential* scene features gives only a low recognition rate of 30.03%, however when introducing features over several time steps $\mathbf{F}_t$, we achieve a higher rate of 35.80%. Adding the confidence values $\mathbf{C}_t$ and the previous tracker identifier $\mathbf{S}_{t-1}$, the rate is considerably increased to 81.80%. Given the low recognition rates without $\mathbf{S}_{t-1}$, one might think that the classifier's decision is mainly relying on this particular feature. However, when training solely on the features $\mathbf{S}_{t-1}$, the classifier does not converge to a viable solution as $\mathbf{S}_{t-1}$ alone does not allow for a good generalisation. It tends to just respond the identifier of the previously selected tracker. In fact, it is the association of both the context scene features and tracker features that enables the classifier to extract and learn the correlations between the scene information and the trackers' performance.

5.2 Tracking evaluation

We further evaluated the performance of the tracking framework and its different components following the protocol of the VOT2013 benchmark [13]. It's a well-known benchmark among the tracking community, which help us compare our proposed framework to the state of the art tracking methods. The tracking algorithm is initialised with the ground truth object

Classifier input $\mathbf{i}_t$	Recognition rate
$\{\mathbf{f}_t^L\}$	26.28%
$\{\mathbf{f}_t^L, \mathbf{f}_t^G\}$	27.25%
$\{\mathbf{f}_t^L, \mathbf{f}_t^G, \mathbf{f}_t^D\}$	30.03%
$\{\mathbf{F}_t\}$	35.80%
$\{\mathbf{F}_t, \mathbf{C}_t\}$	40.06%
$\{\mathbf{F}_t, \mathbf{C}_t, \mathbf{S}_{t-1}\}$	**81.80%**

Table 1: Recognition results for different classifier inputs on the VOT2013 database.

Method	Accuracy	Failures
Best Confidence (BC)	0.559	3.513
Context Classifier	0.540	1.617
Context Classifier with HMM	**0.574**	0.887
Context Classifier with HMM and Kalman Filter	0.553	**0.583**

Table 2: VOT2013 Benchmark results for the proposed method and baseline (BC).

bounding box in the videos' first frame and re-initialised whenever the target is lost. The benchmark provides two evaluation measures: *"accuracy"* is the average overlap with the ground truth, and *"failures"* represents the robustness of the algorithm counting the number of times the tracking is lost. Here, we present the (sequence-based) average values for the whole dataset. See [13] for more details.

In Table 2, the results for our proposed tracking method and its different components are presented along with a baseline method called "Best Confidence" (BC) that selects the best tracker only by the maximum confidence value. A classification based on the global scene context features considerably reduces the number of failures compared to using only the confidence values in BC. We can also see that both the HMM and the Kalman Filter greatly improve the robustness. However, when decreasing the number of failures the accuracy decreases slightly as well. These two additional components of the algorithm are important for the *continuity* of the tracking and, at the same time, ensure that a *different*, more suitable tracker can be selected whenever drastic scene changes occur.

We further compare the proposed framework (*i.e.* including HMM and Kalman Filter) with other state-of-the-art tracking algorithms, as well as the three individual (OAB-based) trackers. Figure 2 shows the ranking and Accuracy-Robustness plots. The proposed algorithm increases the robustness of our individual trackers HAAR, HOG and HOC. In fact, our method ranks among the top trackers of the challenge in terms of robustness, outperforming for example EDFT [7], STC [18], Struck [10] and FoT [17] methods. On the other hand, the accuracy of our method is directly linked and dependent on the accuracy of the individual trackers. Note that we fixed the scale of the trackers in our experiments as the robustness of OAB generally decreased when adapting to scale. Using more accurate trackers would increase the general accuracy of our proposed method as well as the robustness. Nevertheless, we demonstrated that our framework is able to combine the strengths of each OAB tracker and enhance the overall robustness (Fig. 3).

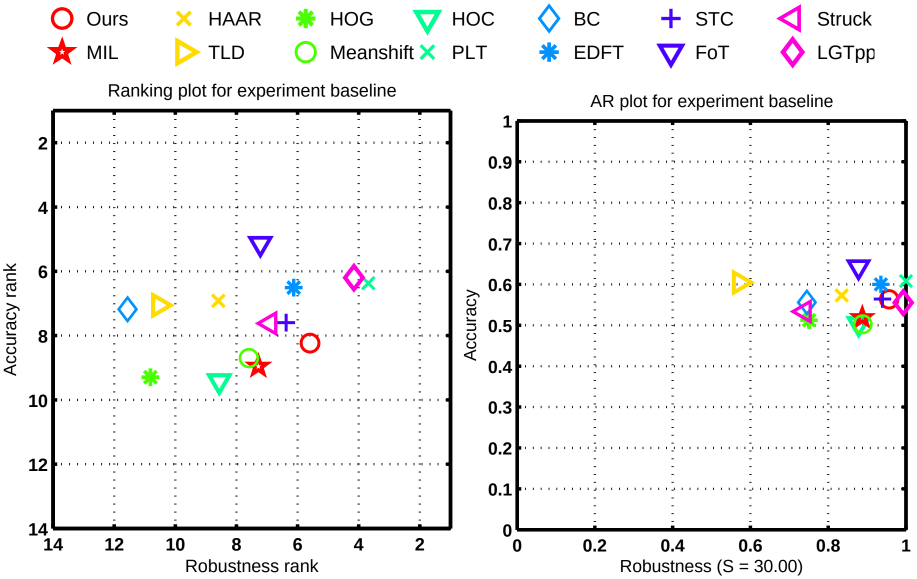

Figure 2: VOT2013 Benchmark ranking (left) and Accuracy-Robustness (right) plots: Comparison of our proposed framework (Ours) with Online AdaBoost trackers HAAR, HOG, HOC ; baseline BC ; and state of the art trackers: STC [18], Struck [10], MIL [1], TLD [12], Meanshift [5], PLT [13], EDFT [7], FoT [17], LGT++ [3].

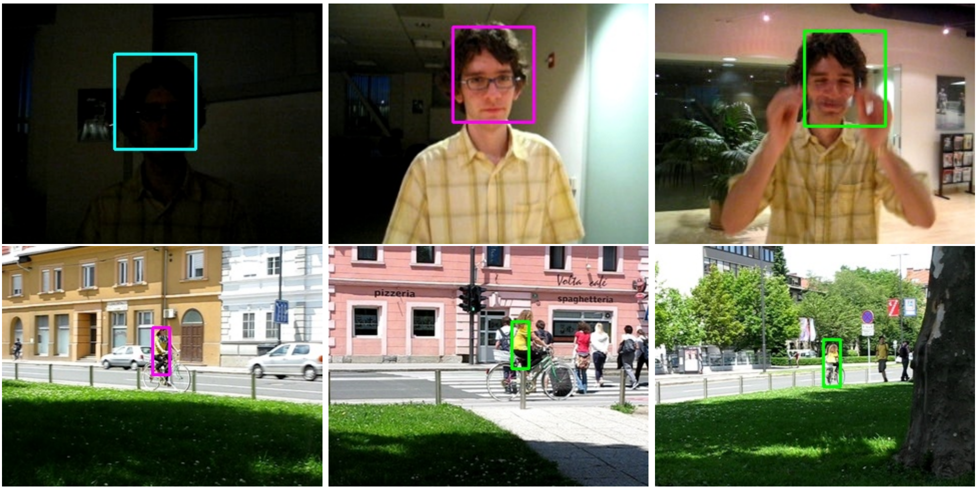

Figure 3: Illustration of our proposed framework's tracking results on the "David"(1^{st} row) and "Bicycle"(2^{nd} row) videos. Different scene context variations in lighting, texture or background are present throughout the videos. Our framework selects the most suitable tracker in each scenario (*pink*: HAAR, *blue*: HOG, *green*: HOC).

6 Conclusion

In this work, we proposed a novel tracker selection framework based on scene context. We used a classifier to learn the patterns that relate the scene context information with the "suitability" of specific independent trackers under the conditions at a given video frame. We also introduced a HMM to eliminate outliers and enforce the continuity in our tracking. In our experiments with the VOT2013 benchmark, the proposed method ranks among the top state-of-the-art trackers, and we showed the effectiveness of generic scene context features in challenging tracking environments. Future work will concentrate on studying the effect of using diversified state of the art trackers, increasing the number of trackers, as well as using other types of scene context features.

References

[1] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[2] Christian Bailer, Alain Pagani, and Didier Stricker. A superior tracking approach: Building a strong tracker through fusion. In *Proceedings of the ECCV*, pages 170–185, 2014.

[3] Luka Cehovin, Matej Kristan, and Ales Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):941–953, 2013.

[4] Robert T. Collins and Yanxi Liu. On-line selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

[5] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[6] TB Dinh, Nam Vo, and G Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Proceedings of BMVC*, 2011.

[7] Michael Felsberg. Enhanced distribution field tracking using channel representations. In *Proceedings of the ICCV (Workshops)*, 2013.

[8] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *Proceedings of BMVC*, pages 47–56, 2006.

[9] Helmut Grabner, Jiri Matas, Luc Van Gool, and Philippe Cattin. Tracking the invisible : Learning where the object might be. In *Proceedings of CVPR*, pages 1285–1292, 2010.

[10] Sam Hare, Amir Saffari, and Philip H.S. Torr. Struck: structured output tracking with kernels. In *Proceedings of the ICCV*, pages 263–270, 2011.

[11] Zhibin Hong, Xue Mei, and Dacheng Tao. Dual-force metric learning for robust distracter-resistant tracker. In *Proceedings of the ECCV*, pages 513–527, 2012.

[12] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2011.

[13] Matej Kristan, Luka Cehovin, Roman Pflugfelder, Georg Nebehay, Gustavo Fernandez, Jiri Matas, and et al. The Visual Object Tracking VOT2013 challenge results. In *Proceedings of the ICCV (Workshops)*, 2013.

[14] Junseok Kwon and K.M. Lee. Visual tracking decomposition. In *Proceedings of CVPR*, pages 1269–1276, 2010.

[15] Junseok Kwon and K.M. Lee. Tracking by sampling trackers. In *Proceedings of the ICCV*, 2011.

[16] Ido Leichter, Michael Lindenbaum, and Ehud Rivlin. A general framework for combining visual trackers – “black boxes” approach. *IJCV*, 67(3):343–363, March 2006.

[17] Jiri Matas and Tomas Vojir. Robustifying the flock of trackers. In *Comp. Vis. Winter Workshop*, 2011.

[18] Salma Moujtahid, Stefan Duffner, and Atilla Baskurt. Coherent selection of independent trackers for real-time object tracking. In *VISAPP*, 2015.

[19] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[20] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of IEEE*, 92(3):495–513, 2004.

[21] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007.

[22] Shuran Song and Jianxiong Xiao. Tracking revisited using RGBD camera: Unified benchmark and baselines. In *Proceedings of the ICCV*, 2013.

[23] Severin Stalder, Helmut Grabner, and L Van Gool. 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, 2009.

[24] Björn Stenger, Thomas Woodley, and Roberto Cipolla. Learning to track with multiple observers. In *Proceedings of CVPR*, 2009.

[25] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the ICCV*, pages 1023–1029, 2003.

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

[27] Longyin Wen, Zhaowei Cai, Zhen Lei, Dong Yi, and Stan Z. Li. Robust online learned spatio-temporal context model for visual tracking. *IEEE Transactions on Image Processing*, 23(2):785–796, 2014.

[28] Ming Yang, Ying Wu, and Gang Hua. Context-aware visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1195–1209, July 2009.

[29] Alper Yilmaz, Xin Li, and Mubarak Shah. Object contour tracking using level sets. In *Proceedings of the ACCV*, 2004.

[30] Zhaozheng Yin, Fatih Porikli, and Robert T. Collins. Likelihood map fusion for visual object tracking. In *IEEE Workshop on Applications of Computer Vision*, pages 1–7, January 2008.