Online Feature Selection for Visual Tracking

Giorgio Roffo
giorgio.roffo@univr.it
Simone Melzi
simone.melzi@univr.it

Department of Computer Science
University of Verona
Verona, Italy

Abstract

Object tracking is one of the most important tasks in many applications of computer vision. Many tracking methods use a fixed set of features ignoring that appearance of a target object may change drastically due to intrinsic and extrinsic factors. The ability to dynamically identify discriminative features would help in handling the appearance variability by improving tracking performance. The contribution of this work is threefold. Firstly, this paper presents a collection of several modern feature selection approaches selected among filter, embedded, and wrapper methods. Secondly, we provide extensive tests regarding the classification task intended to explore the strengths and weaknesses of the proposed methods with the goal to identify the right candidates for online tracking. Finally, we show how feature selection mechanisms can be successfully employed for ranking the features used by a tracking system, maintaining high frame rates. In particular, feature selection mounted on the Adaptive Color Tracking (ACT) system operates at over 110 FPS. This work demonstrates the importance of feature selection in online and realtime applications, resulted in what is clearly a very impressive performance, our solutions improve by 3% up to 7% the baseline ACT while providing superior results compared to 29 state-of-the-art tracking methods.

1 Introduction

The last decade has seen a revolution in the theory and application of computer vision. Many powerful algorithms for visual tracking have been proposed [1, 9, 16, 18]. Most of the tracking systems employ a set of features which are used statically across successive frames [17]. It is well known that tracking deals with image streams that change over time [7], therefore, data will easily produce frames in which the object has low-contrast or the target region is blurred (e.g. due to the motion of the target or camera), in these cases to obtain robust tracking a great deal of prior knowledge about scene structure or expected motion is imposed [3, 34, 38], and thus tracking success is bought at the price of reduced generality. Selecting the right features plays a critical role in tracking [6, 7, 12, 31]. Trackers which comprises feature selection strategies into their inner engines can be flexible enough to handle gradual changes in appearance in real scenarios [9]. The degree to which a tracker can discriminate a target from its surroundings is directly related to this flexibility. Since foreground and background appearance can also drastically change as the target object moves from place to place, tracking features should also need to adapt.

Many feature selection methods used in off-line settings (e.g., bioinformatics, data mining [14, 26, 28, 36]) have been so far largely neglected, to the best of our knowledge, at the

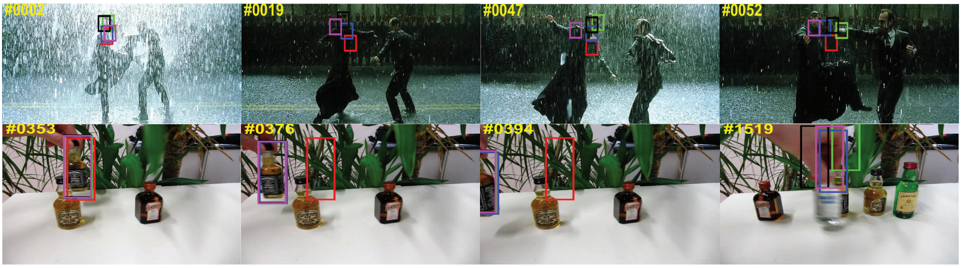

Figure 1: Comparison of feature selection approaches embedded on the ACT tracker in challenging situations such as fast motion, out-of-view, occlusion, illumination variation, motion blur. The results of ACT, ACTFisher, ACTMI, ACT-mRMR and ACTinffs are represented by red, green, blue, black and pink boxes respectively.

level of online visual tracking, especially under the hard constraint of speed required for target tracking. The contribution of this work is threefold. Firstly, we propose a pool of modern feature selection algorithms (among filter, embedded, and wrapper methods) selected so as to meet the requirements of a real-time application. Secondly, we investigate the strengths and weaknesses of these methods for a classification task in order to identify the right candidates for visual tracking. Finally, the selected candidates are embedded on the Adaptive Color Tracking system [9] (ACT). We extensively test the solutions on 50 test sequences from the Online Object Tracking [35] (OTB-50) benchmark. Our solutions' performance has been evaluated with the same protocol and methodology provided by the OTB-50 benchmark. The baseline ACT and its variants, with different feature selection mechanisms, have been also compared against 29 state-of-the-art trackers (it is worth noting that 10 of them use feature selection as part of their framework).

This paper demonstrates the importance of feature selection in online and realtime applications, resulted in what is clearly a very impressive performance. Figure 1.**A** presents tracking results in an environment of illumination variations, occlusions, out-of-views, fast motion among other challenging aspects. The example frames are from the *Matrix* and *liquor* sequences. All the feature selection approaches performs favorably against the baseline ACT tracker (red box).

The rest of the paper is organized as follows: Section 2 presents a brief look at online feature selection for visual tracking, mostly focusing on the comparative approaches we consider in this study. In Section 3 we present the ACT tracker, and in Section 4 how we embedded feature selection in it. Section 5 presents the proposed set of feature selection methods and the computational complexity is given. Extensive experiments are reported in Section 6. In Section 6.1 we analyze the strengths and weaknesses of the proposed method regarding the classification task. In Section 6.2 we introduce the dataset, define the evaluation methodology and report results. Finally, in Section 7, conclusions are given.

2 Related Work

The problem of real-time feature selection has, to the best of our knowledge, rarely been addressed in the literature, especially under the hard constraint of speed required for target tracking.This section presents the related literatures of real-time feature selection for target tracking, mainly focusing on the comparative approaches used in the experimental section.

The Variance Ratio (VR) [7] tracker is an online feature ranking mechanism based on applying the two-class variance ratio to log likelihood distributions computed for a given feature from samples of object and background pixels. This feature ranking approach is embedded in a tracking system that selects top-ranked features for tracking. Other recent variants of VR are provided by the Video Verification of Identity (VIVID) testbed [6] which includes the mean shift (MS-V), template matching (TM-V), ratio shift (RS-V), and peak difference (PD-V) methods. Another robust tracker is the Sparsity-based Collaborative Model [39] (SCM), it is an object tracking algorithm which uses a robust appearance model that exploits both holistic templates and local representations, it is based on the collaboration of generative and discriminative modules where feature selection is part of the framework. Moreover, the On-line AdaBoost (OAB) [12] tracker, by using fast computable features like Haar-like wavelets, orientation histograms, local binary patterns, selects the most discriminating features for tracking resulting in stable tracking results. OAB method does both - adjusting to the variations in appearance during tracking and selecting suitable features which can learn object appearance and can discriminate it from the surrounding background.

Semi Boosting Tracker (SBT) and Beyond Semi Boosting Tracker (BSBT) [31] are multiple object tracking approaches which extend semi-supervised tracking by object specific and adaptive priors. Valuable information which would be ignored in a pure semi-supervised approach is safely included in the prior using a detector for validation and a tracker for sampling. The prior is interpreted as recognizer of the object as similar objects are distinguished. The tracked objects are used to train local detectors to simplify detection in the specific scene. By using feature selection the classifier framework is able to track various objects, even under appearance changes and partial occlusions, in challenging environments.

Finally, the Scale of the Mean-Shift [5] (SMS) is another efficient technique for tracking 2D blobs through an image. Lindeberg's theory of feature scale selection based on local maxima of differential scale-space filters is applied to the problem of selecting kernel scale for mean-shift blob tracking.

3 The Adaptive Color Tracking System

The ACT system [9] is one of the most recent solutions for tracking, which extends the CSK tracker [17] with color information. ACT exploits color naming (CNs), proposed in [33] (i.e., the action of assigning linguistic color labels to image pixels), to target objects and learn an adaptive correlation filter by mapping multi-channel features into a Gaussian kernel space. Schematically, the ACT tracker contains three improvements to CSK tracker: (i) A temporally consistent scheme for updating the tracking model is applied instead of training the classifier separately on single samples, (ii) CNs are applied for image representation, and (iii) ACT employs a dynamically adaptive scheme for selecting the most important combinations of colors for tracking.

In the ACT framework, for the current frame p, CNs are extracted and used as features for visual tracking. Moreover, a grayscale image patch is preprocessed by multiplying it with a Hann window [9], then, the final representation is obtained by stacking the luminance and color channels. The ACT algorithm considers all the extracted appearances x^p to estimate the associated covariance matrix C_p. A projection matrix B_p, with orthonormal column vectors, is calculated by eigenvalue decomposition (EVD) over C_P. Let x_1^p be the D_1-dimensional learned appearance. Thus, the projection matrix B_p is used to compute the new D_2-dimensional feature map x_2^p of the appearance by the linear mapping $x_1^p(m,n) = B_p^T x_2^p(m,n), \forall m,n$. The projection matrix B_p is updated by selecting the D_2 normalized eigenvectors of R_p (see Eq.1), that corresponds to the largest eigenvalues.

$$R_p = C_p + \sum_{j=1}^{p-1} B_j \Lambda_j B_j^T, \tag{1}$$

where C_p is the covariance matrix of the current appearance and Λ_j is a $D_2 \times D_2$ diagonal matrix of weights needed for each basis in B_p. Finally, D_2 represents the number of dimensions where the D_1 features are projected on.

Summarizing, B_p is used to transform the original feature space to yield a subspace by performing dimensionality reduction. Finally, the projected features are used to compute the detection scores and the target position in in the new frame $p+1$ (see [9, 17] for further details).

4 Embedding Feature Selection

We present a collection of modern algorithms in Sec. 5, suitable to be embedded on the ACT system. For all of them, the following steps are taken in the embedding procedure. Firstly, the ACT appearances x are extracted for object and background classes, and computed using samples taken from the most recently tracked frame. Secondly, feature ranking is applied in a supervised/unsupervised manner depending on the selection method. This important step can be interpreted as ranking by relevance the dimensions of the feature vector x (10-D), where features in the first ranked positions are tailored to the task of discriminating object from background in the current frame (note: in Section 6.2 the first 4 features have been selected). Finally, the most discriminative features are used to estimate the C covariance matrix used to feed the ACT. This procedure enables ACT to continually update the set of features used, which turns out to better separate the target from its immediate surroundings.

5 Feature Selection Methods

In this work we propose a pool of modern feature selection methods with the goal in mind to choose some good candidates among them for a final application in real-time visual tracking. *Relief-F* [23] is a widely used randomized, iterative, and supervised approach that computes the quality of the features according to how well their values differentiate data samples that are near to each other; it does not take into account the redundancy of features, and performance decreases with few data. Similar problems affect SVM-RFE [15], which is an embedded method that selects features in a sequential, backward elimination manner, ranking high a feature if it strongly separates the samples by means of a linear SVM (see Table 1 in Section. 5 for further details). An effective yet fast filter method is the *Fisher* method [14], it computes a score for a feature as the ratio of interclass separation and intraclass variance, where features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones. Other widely used filters are based on mutual information, such as *MutInf* [36] and mRMR [26]. The former considers as a selection criterion the mutual information between the distribution of the values of a given feature and the membership to a particular class.The latter employs the minimal-redundancy maximal-relevance criterion, mRMR [26]. mRMR is an efficient incremental search algorithm. Relevance scores are assigned by maximizing the joint mutual information between the class variables and the subset of selected features. The computation of the information between high-dimensional

Acronym	Type	Cl.	Compl.
SVM-RFE [15]	e	s	$\mathcal{O}(T^2 n log_2 n)$
Inf-FS [28]	f	u	$\mathcal{O}(n^{2.37}(1+T))$
Relief-F [23]	f	s	$\mathcal{O}(iTnC)$
FSV [4, 13]	w	s	N/A
MutInf [36]	f	s	$\sim \mathcal{O}(n^2 T^2)$
mRMR [26]	f	s	$\mathcal{O}(n^3 T^2)$
Fisher [14]	f	s	$\mathcal{O}(Tn)$

Table 1: List of the feature selection approaches considered in the experiments, specified according to their *Type*, class (*Cl.*), and complexity (*Compl.*). As for the complexity, T is the number of samples, n is the number of initial features, i is the number of iterations in the case of iterative algorithms, and C is the number of classes. The complexity of FSV cannot be specified since it is a wrapper (it depends on the chosen classifier).

vectors is impractical, as the time required becomes prohibitive. To face this problem the mRMR propose to estimate the mutual information for continuous variables using Parzen Gaussian windows. This estimate is based on a heuristic framework to minimize redundancy and uses a series of intuitive measures of relevance and redundancy to select features. We also compare our approach against the graph-based method Inf-FS [28]. The inf-FS is an unsupervised filter method. In the Inf-FS formulation, each feature is a node in the graph, weights are given by mixture of correlation and standard deviation between feature distributions. A path over the graph is seen as a possible selection of features. A path integral process is then applied and the higher the final score, the most important (or most different) the feature. As a result, this method assigns a score of “importance" to each feature by taking into account all the possible feature subsets as paths on a graph. Table 1 lists the proposed methods, reporting their *type*, that is, f = filters, w = wrappers, e = embedded methods, and their *class*, that is, s = supervised or u = unsupervised (using or not using the labels associated with the training samples in the ranking operation). Additionally, we report their computational complexity (if it is documented in the literature).

6 Experiments

6.1 Experiment 1: Classification Task

This section proposes a set of tests on the PASCAL VOC-2007 [11] dataset. In object recognition VOC-2007 is a suitable tool for testing models, therefore, we use it as reference benchmark to assess the strengths and weaknesses of using feature selection approaches regarding the classification task while taking care at their execution times. For this reason, we compare seven modern approaches where their details are reported in Table 1.

This experiment considers as features the cues extracted with a deep convolutional neural network architecture (CNN). We selected the recent pre-trained model called GoogLeNet [32], which achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). We use the 1,024-dimension activations of the last five layers as image descriptors (5,120 features in total). The VOC-2007 edition contains about 10,000 images split into train, validation, and test sets, and labeled with twenty object classes. A one-vs-rest SVM classifier for each class is learnt and evaluated independently and the performance is measured as mean Average Precision (mAP) across all classes.

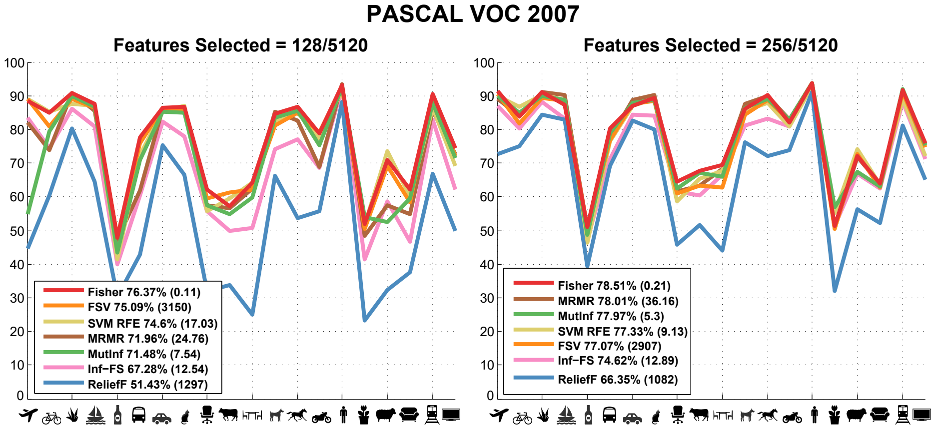

Figure 2: Varying the cardinality of the selected features. The image classification results achieved in terms of mean average precision (mAP). The execution times for each method (in seconds) are reported within the brackets, while selecting the first 128 and 256 features from the total $5,120$.

Figure 2 serves to analyze and empirically clarify how well important features are ranked high by several feature selection algorithms, selecting 128 features on the left and 256 features on the right from the 5,120 available. We report as curves the performance in terms of mean average precision (mAP) on the twenty object classes represented by intuitive icons on the abscissa axis. The legends report the mAP in percentage and the average execution time for each method. It is worth noting that Fisher method achieved the best performance (77.44% as mean mAP on the two tests) followed by the wrapper method FSV (76.08%) and the embedded SVM-RFE (75.97%). The quality of the ranking is compared with the execution time of each method. The average time spent by each method to produce the output for each class shows the supremacy of the Fisher approach which thanks to its low computational complexity free of hidden constants completes the task in less than a second (0.16 as mean time on the two tests). The fastest approaches after Fisher, are MutInf (6.42), inf-FS (12.72), SVM-RFE (13.08), and mRMR (30.46) that are all under the minute while ReliefF $(1,189.50)$ and FSV $(3,028.50)$ are not comparable in terms of time.

6.2 Experiment 2: Feature selection for single object tracking

Taking advantage from the results reported in the previous section, we decided to use the following four candidate methods: *MutInf*, *Fisher*, *Inf-FS*, and *mRMR*. In particular, we take care that execution times of these methods meet the requirements for a real-time application. What is remarkable is that most of these methods achieved good results in terms of average precision for classification and may be examples worth using in tracking. We discarded SVM-RFE because it becomes unstable at some values of the feature filter-out factor [24], i.e., the number of features eliminated in each iteration. We selected the mRMR since in embedded methods, the classifier characteristics such as SVM weights in SVM-RFE provide a criterion to rank features based on their relevancy, but they do not account for the redundancy among the features, while mRMR takes into account both relevancy and redundancy. As a result, our pool consists of 4 filter methods which evaluate a feature subset by looking at the intrinsic characteristics of data with respect to class labels and do not incorporate classifier

Attr	Description
1	Illumination Variation, the illumination in the target region is significantly changed.
2	Scale Variation, the ratio of the bounding boxes of the first frame and the current frame is out of the range $[1/t_s, t_s]$, $t_s > 1$ (t_s=2).
3	Occlusion, the target is partially or fully occluded.
4	Deformation, non-rigid object deformation.
5	Motion Blur, the target region is blurred due to the motion of target or camera.
6	Fast Motion, the motion of the ground truth is larger than tm pixels (tm=20).
7	In-Plane Rotation, the target rotates in the image plane.
8	Out-of-Plane Rotation, the target rotates out of the image plane.
9	Out-of-View, some portion of the target leaves the view.
10	Background Clutters, the background near the target has the similar color or texture as the target.
11	Low Resolution, the number of pixels inside the ground-truth bounding box is less than tr (tr =400).

Table 2: The OTB-50 benchmark includes 50 test sequences from recent literatures and 11 attributes, which represents the challenging aspects in visual tracking (table from [35]).

operation into the selection process.

In line with Section 4, we embedded these methods on the ACT system obtaining four variants: ACTMI for MutInf, ACTFisher, ACTinffs, and ACT-mRMR. We compare them against the baseline ACT, and also against 29 different state-of-the-art trackers shown to provide excellent results in literature. Some trackers include feature selection within their framework such as the VIVID tracker suite (VR-V, PD-V, RS-V, MS-V, and TM-V), SCM, OAB, SBT and BSBT, and SMS (see Sec. 2 for further details). Other trackers used for the comparison are the following: CSK [17], CPF [27], Struck [16], CXT [10], VTD [20], VTS [21], LSK [22], KMS [8], Frag [1], MIL [2], CT [37], TLD [19], IVT [29], DFT [30], ASLA [18], L1APG [3], ORIA [34], MTT [38], and LOT [25].

6.3 Datasets

The OTB-50 benchmark is a tracking dataset with 50 fully annotated sequences to facilitate tracking evaluation. The sequences used in our experiments pose challenging situations such as motion blur, illumination changes, scale variation, heavy occlusions, in-plane and out-plane rotations, deformation, out of view, background clutter and low resolution (see Table 2).

6.4 Results

Generally, trackers evaluation performance is done by running them throughout a test sequence with the initialization from the ground truth position in the first frame and reporting the average precision and success rate. Therefore, we use the same evaluation methodology used in OTB-50, where precision and success rate have been used for quantitative analysis. The former, is a widely used metric, it is the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truth. As a result, a precision plot shows the percentage of frames whose estimated locations is within the given threshold distance of the ground truth. The latter, success rate, measures the bounding box overlap between the tracked bounding box r_t and the ground truth bounding box r_a. The success plot shows the ratio of successful frames as the thresholds varied from 0 to 1. When we consider the initialization from the ground truth in the first frame we refer this as one-pass evaluation

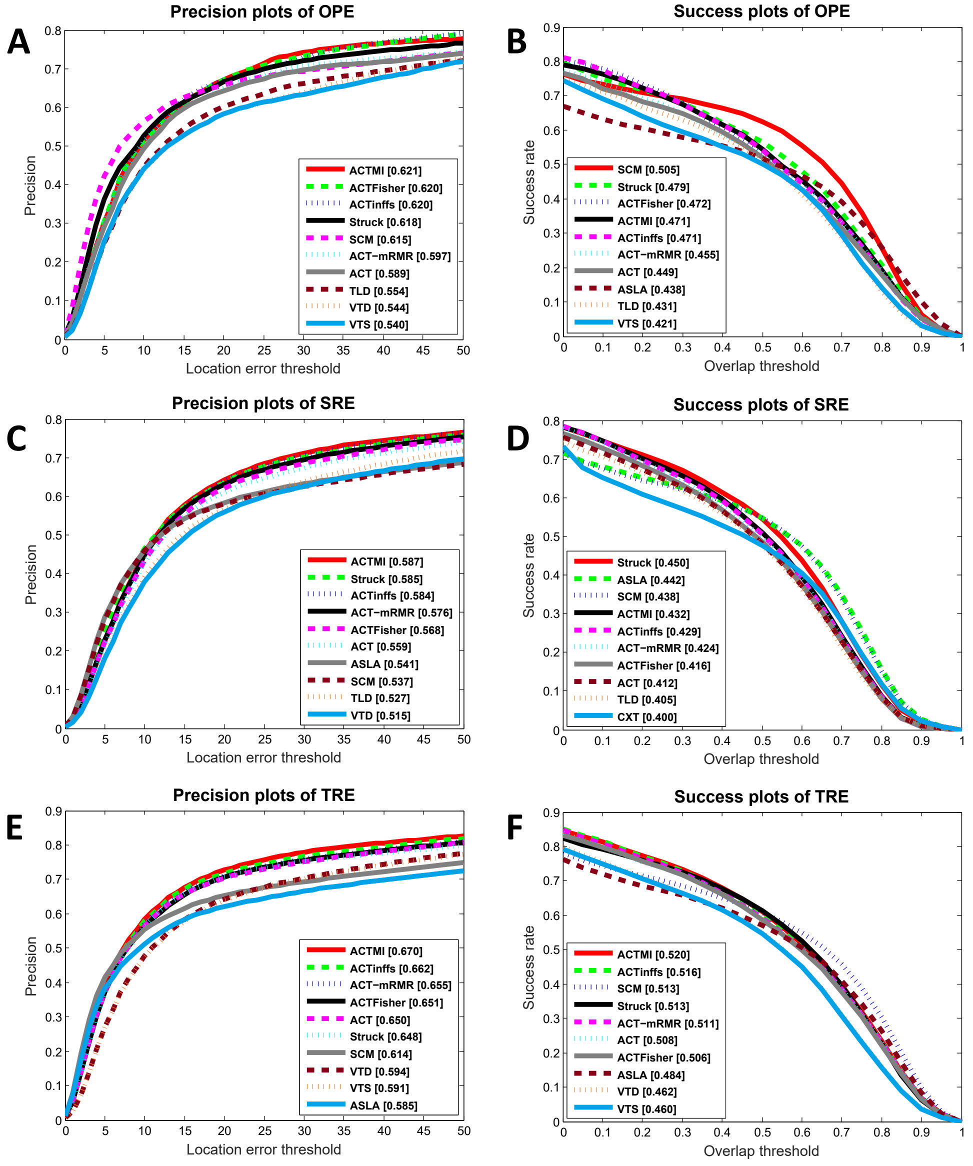

Figure 3: Precision and success plots over all 50 sequences provided by the OTB-50 benchmark. Only the top 10 (out of 34) trackers are displayed for clarity. The mean precision and success scores for each tracker are reported in the legend.

(OPE). Figure 3.(A-B) shows precision and success plots for OPE criterion.

Note that ACTMI, ACTFisher and ACTInfFS improve the baseline ACT tracker more than 3% in mean distance precision, and more than 2% on bounding box overlaps. Feature selection methods used in this work perform variable rankings, for such a reason we decided to reduce the problem dimensionality by 60%, so the amount of selected features is up to 4. Since trackers may be sensitive to the initialization, in [35] other two ways to analyze

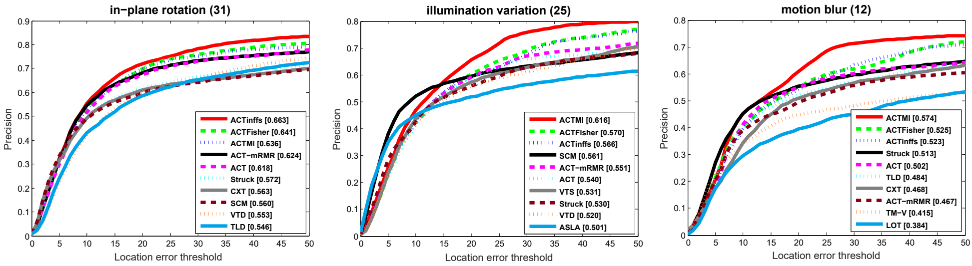

Figure 4: This figure presents three challenging tracking problems that illustrate the benefits of combining online feature selection with object tracking.

a tracker's robustness to initialization have been proposed. These tests are referred as temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). As for SRE, the initial bounding box in the first frame is sample by shifting or scaling the ground truth. Figure 3.(C-D) reports tracking results for the SRE criterion, we used 8 spatial shifts and 4 scale variations. Thus we evaluated each tracker 12 times for SRE. In TRE tests, the tracker is initialized at different t time instances. At each time, the initial bounding box location is set according with the ground truth, then the tracker run to the end of the sequence. We applied perturbations at 20 different times to each tracker TRE. Figure 3.(E) shows how ACTMI improves the baseline ACT by 2% in mean distance precision. In both cases, Figure 3.(E-F), ACTMI, ACTinffs, and ACT-mRMR perform favorably to the baseline ACT. Noteworthy, averaging precision scores across the three criteria, ACTMI (62.6% / 19.0 fps) define in general the new top-score on this benchmark, followed by ACTinffs (62.2% / 111.4 fps) where both overcome the ACT (59.9% , 196 fps). As for the other trackers, SCM (58.9% / 0.5 fps) and Struck (61.7% / 20.2). As a result, ACTinffs turns out to be the best trade-off between accuracy (62.2%) and speed (111.4 fps). We say ACTinffs has the same order of magnitude of the ACT in terms of fps.

Figure 4 shows example precision plots of different attributes. Note, only the top 10 trackers out of 34 are displayed. From left to right: in-plane rotation (comprises 31 sequences), illumination variations (25 sequences) and motion blur (12 sequences). As for in-plane rotation, Figure 4 shows a scenario where the ACT overcomes the competitors, in such a case feature selection allows to improve its performance by 4.5%. Illumination variation and motion blur scenarios represent a more challenging task, where the appearance of the target changes drastically due to extrinsic factors, the ability to identify the most discriminative features for the current frame allows to obtain an improvement in precision up to 7.6% and 7.2% respectively, and at the same time to overcome the methods in comparison.

For all the attributes the ACTMI, ACT-Fisher, and ACTinffs provide superior results compared to existing methods. This is due to the fact that feature selection allows a photometric invariance while preserving discriminative power. Even in those situations where the baseline ACT does not perform top performance, feature selection permits to overcome the other methods (e.g., for motion blur ACTMI improves the ACT by 7% in precision).

7 Conclusions

In this paper we evaluated a collection of seven modern feature selection approaches, used in off-line settings so far. We investigated the strengths and weaknesses of these algorithms in a classification setting to identify the right candidates for a real-time task. We selected four candidates who meet the requirements of speed and accuracy for visual tracking. Finally, we showed how these feature selection mechanisms can be successfully used for ranking features combined with the ACT system, and, at the same time, maintaining high frame rates (ACTinffs operates at over 110 FPS). Results show that our solutions improve by 3% up to 7% their baseline. Moreover, ACTMI resulted in a very impressive performance in precision, providing superior results compared to 29 state-of-the-art tracking methods. We hope that this work motivates researchers to take into account the use of fast feature selection methods as an integral part of their tracker systems. For the sake of repeatability, a code library will be posted on the project page to provide the material needed to replicate our experiments.

References

[1] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[2] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.

[3] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[4] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Conf. International Conference on Machine Learning (ICML)*, 1998.

[5] Robert Collins. Mean-shift blob tracking through scale space. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2003.

[6] Robert Collins, Xuhui Zhou, and Seng Keat Teh. An open source tracking testbed and evaluation web site. In *IEEE Workshop Performance Evaluation of Tracking and Surveillance (PETS)*, 2005.

[7] Robert T. Collins, Yanxi Liu, and Marius Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005.

[8] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2003.

[9] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost van de Weijer. Adaptive color attributes for real-time visual tracking. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] Thang Ba Dinh, Nam Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[12] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Conf. British Machine Vision Conference (BMVC)*, 2006.

[13] Guillermo L. Grinblat, Javier Izetta, and Pablo M. Granitto. SVM based feature selection: Why are we using the dual? In *Conf. Ibero-American Conference on AI*, 2010.

[14] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *Computing Research Repository (CoRR)*, 2012.

[15] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning Journal*, 2002.

[16] Sam Hare, Amir Saffari, and Philip H. S. Torr. Struck: Structured output tracking with kernels. In *IEEE Conf. International Conference on Computer Vision (ICCV)*, 2011.

[17] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Conf. European Conference on Computer Vision (ECCV)*, 2012.

[18] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[20] Junseok Kwon and Kyoung Mu Lee. Visual tracking decomposition. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[21] Junseok Kwon and Kyoung Mu Lee. Tracking by sampling trackers. In *IEEE Conf. International Conference on Computer Vision (ICCV)*, 2011.

[22] Baiyang Liu, Junzhou Huang, Lin Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[23] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman & Hall Data Mining and Knowledge Discovery Series, 2008.

[24] Piyushkumar A Mundra and Jagath C Rajapakse. SVM-RFE with mRMR filter for gene selection. *IEEE Transactions on Nanobioscience*, 2010.

[25] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. Locally orderless tracking. *International Journal of Computer Vision*, 111(2):213–228, 2015.

[26] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005.

[27] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Conf. European Conference on Computer Vision (ECCV)*, 2002.

[28] Giorgio Roffo, Simone Melzi, and Marco Cristani. Infinite feature selection. In *IEEE Conf. International Conference on Computer Vision (ICCV)*, 2015.

[29] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 2008.

[30] Laura Sevilla-Lara and Erik Learned-Miller. Distribution fields for tracking. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[31] Severin Stalder, Helmut Grabner, and Luc Van Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *IEEE Conf. International Conference on Computer Vision (ICCV)*, 2009.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] Joost van de Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 2009.

[34] Yi Wu, Bin Shen, and Haibin Ling. Online robust image alignment via iterative convex optimization. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[36] Marco Zaffalon and Marcus Hutter. Robust feature selection using distributions of mutual information. In *Conf. International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[37] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking. In *Conf. European Conference on Computer Vision (ECCV)*, 2012.

[38] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[39] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang. Robust object tracking via sparsity-based collaborative model. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.