Not All Negatives are Equal: Learning to Track with Multiple Background Clusters

Gao Zhu, *Student Member, IEEE,* Fatih Porikli, *Fellow, IEEE,* and Hongdong Li, *Member, IEEE*

***Abstract*—Conventional tracking-by-detection approaches for visual object tracking often assume that the task at hand is a binary foreground-versus-background classification problem where the background is a single, generic, and all-inclusive class. In contrast, here we argue that the background appearance, for the most part, possesses a more complicated structure that would benefit from further partitioning into multiple contextual clusters. Our observation is that, although the background class is contemplated to contain a vast intra-class variation, during the tracking process only a small portion of this diversity is present at the current frame around the foreground object. This observation motivates us to build multiple fine-grained foreground-versus-contextual-cluster models that provide more discriminative classifications, and consequently more robust and accurate foreground object tracking. For each cluster, we employ a structured output support vector machine (SSVM), and in an online manner, we combine the responses of multiple classifiers. To this end, we apply a top level SSVM that models the tracked foreground object. We show that our refined modeling of the background is better than naively growing the complexity of a single foreground-background classifier, i.e. increasing the number of support vectors that existing approaches rely on, which cause over-fitting issues. Our extensive evaluations on large benchmark datasets demonstrate that our tracker consistently outperforms the current state-of-the-art while having comparable computational requirements.**

***Index Terms*—Tracking-by-detection, contextual clustering, fine-grained model, support vector machine (SVM).**

I. Introduction

VISUAL object tracking confronts with major challenges due to object appearance and scene illumination variation, partial and full occlusion, background clutter, and noise. To build a tracker that is robust to such issues, tracking-by-detection techniques learn adaptive object models, e.g. classifiers, in an online fashion and then search for the best match in the consecutive frames.

Depending on the basic learning strategy, tracking-by-detection approaches can be grouped into generative and discriminative learning categories. Generative learning based models mainly concentrate on how to construct an object representation in specific feature spaces, including the subspace learning [1], sparse representation [2], [3], [4] and so on [5]. A known drawback of these methods is that they often ignore the influence of the background, and consequently suffer from distractions caused by the background regions with similar appearance to the foreground object. In contrast, discriminative learning based appearance models aim to maximize the inter-class separability between the object and background regions using discriminative learning techniques, including SVMs [6], [7], [8], [9], [10], random forest [11], and multiple instance learning [12], to name a few. Among the main challenges of discriminative methods one can consider how to maintain positive and negative training samples, and how to build a powerful classifier out of them. As the number of the processed frames increases, the number of positive, and in particular negative, samples could inflate. Thus, the design of an adequate model update strategy for discriminative learning is not trivial.

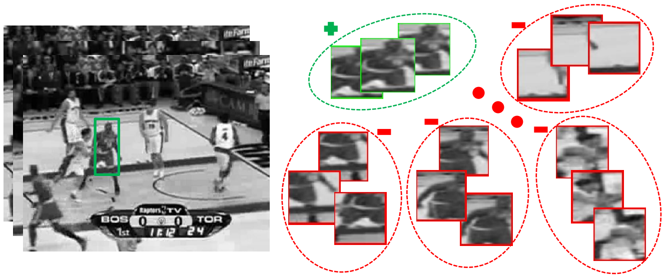


Fig. 1: Instead of training only one classifier to separate the set of positive samples and the set of negative samples, this paper explores the implicit data structure underneath the negative samples by fine-grain partitioning the negatives into multiple clusters. Note that, each background cluster is meaningful, either corresponding to the shifted-versions of the object or further away yet visually similar negative samples of *pure* background samples.

In this paper, we propose to exploit the underlying distribution structure of the training samples to reduce the burden of the classification task, hence increase the discriminative power of the object tracker. To this end, we utilize the weak visual structure of background, i.e., the negative sample space, by explicitly grouping background samples into multiple contextual clusters. Here, a contextual cluster means a group of samples that exhibit similar visual properties and possibly spatial proximity as dissected more in the experimental section.

We observe that these contextual clusters emerge mainly as two distinct groups: the shifted-versions of the object

This work was supported in part by the Australian Research Council (ARC) under Grant DP150104645, DP120103896 and ARC Centre of Excellence CE140100016.

G. Zhu is with the Research School of Engineering, Australian National University. (e-mail: gao.zhu@anu.edu.au).

F. Porikli is with the Research School of Engineering, Australian National University, Data61/CSIRO, and Australian Centre for Robotic Vision. (e-mail: fatih.porikli@anu.edu.au).

H. Li is with the Research School of Engineering, Australian National University, Data61/CSIRO, and Australian Centre for Robotic Vision. (e-mail: hongdong.li@anu.edu.au).

window, which help better localization albeit cause confusion or drift if not modeled properly, and ordinary non-object like background samples, which encourage better detection yet can cause sudden jumps in the subsequent frames if neglected. We show that explicitly building multiple foreground-versus-cluster classifiers increases the discriminative power by preventing object window drifts and avoiding inaccurate assignments. In other words, using multiple background models is preferable to employing only one.

We exploit structured output support vector machines (SSVM) to obtain an individual tracker for each contextual object-cluster pair. First, we train independently each classifier with their respective contextual clusters using low-level feature descriptors such as histogram of intensity. Then, a unifying SSVM is constructed with all negative samples to learn the importance weights corresponding to each contextual SSVM, by concatenating their responses of the training sample into a feature vector. This lends itself to naturally and optimally combining the outcomes from multiple trackers. More details can be found in Section III-B.

On the TB50 dataset [13], our method improves the *precision score* around 11.3% overall. For specific attributes, the performance improvement is up to 23.5% for the deformation case, 17.0% for the fast motion case, 23.7% for the motion blur case, 12.2% for the occlusion, and 7.6% for the background clutter case in comparison to the baseline tracker that uses a single background model. Similarly, our results on the popular VOT2014 [14] and OTB datasets [15] are superior to the baseline tracker by a significant margin.

II. Related Work

For completeness, we provide a brief overview of the most relevant works and refer readers to the object tracking surveys [15], [16], [17].

Among notable approaches, Avidan [6] proposed an SVM-based tracking-by-detection algorithm for distinguishing the object from its close neighborhood. Tian *et al.* [18] utilized the ensemble version of the linear SVM classifiers that can be weighted according to their discriminative abilities at each frame. Henriques *et al.* [19] addressed the high redundancy of the negative samples due to overlapping pixels with circulant matrix and diagonalized it with the Discrete Fourier Transform, reducing both storage and computation by several orders of magnitude. Li *et al.*[20] partitioned the entire image sequence into spatially and temporally adjacent sub-sequences. They then trained an SVM classifier for object/non-object classification on each of these sub-sequences. A spatiotemporal weighted Dempster-Shafer scheme was presented to combine the discriminative information from these classifiers. Nevertheless, none of these algorithms consider the available contextual information as we do.

Recently, deep convolutional neural network (CNN) based solutions have achieved significant advances in object detection and classification tasks [21]. For visual object tracking, [22] employs a candidate pool of multiple CNNs as a data-driven model of different instances of the target object. Inspired by this, [23] interprets the hierarchies of convolutional layers as a nonlinear counterpart of an image pyramid representation and adaptively learns correlation filters on each convolutional layer to encode the target appearance. The recent work in [24] pre-trains a CNN using a large set of videos with ground truth trajectories. The network is composed of shared layers and multiple branches of domain-specific layers and trained with respect to each domain iteratively to obtain generic target representations in the shared layers. However, utilizing the CNN framework to explore the contextual information for a tracker remains an open problem.

Towards incorporating larger receptive fields, Yang *et al.* [25] proposed a context-aware tracking algorithm that considers a set of auxiliary objects as the context of the foreground. These auxiliary objects need to satisfy conditions such as persistent co-occurrence with the foreground and consistent motion correlation. These conditions may not be easily satisfied in practice. [26], [27], [28] used similar concepts termed as ‘distracters’ and ‘supporters’. Distracters [27], [28] are regions that have similar appearance as the target, and supporters [26], [27] are regions or features around the target with consistent co-occurrence and motion correlation in a short time span. These methods require careful maintaining models for distracters and supporters. Li *et al.* [7], showed that the high-order contextual information from samples can increase the robustness of the classifier to noise. The high-order context is defined as a group of samples having some common properties. Each sample in the high-order context is influenced by other samples in the same high-order context. For their tracker, the similarity measure depends on not only two individual samples but also their corresponding contexts. Even though the high-order context provides complementary information to counteract the impact of noise, it still lacks a mechanism to incorporate background context.

The idea of splitting the data into groups and to train a separate classifier for each group to handle the large intra-class variability is proved to be successful, mainly based on boosting algorithms in image classification and object detection [29], [30], [31], [32]. In particular, Godec *et al.* [31] introduced a set of virtual classes generated by a context-driven clustering to cope with the intra-class variability in object detection. They used an online multi-class classifier to initiate and update new virtual classes, and then label a given patch by one of the virtual classes.

Our method does not require explicit labeling of the background into multiple classes. Instead of using a multi-class structure, our tracker operates in a more efficient and consistent manner when it maintains the set of object-versus-contextual clusters. Thus, it is a binary labeling scheme. In addition, not having to explicitly label for multiple background classes enables our method to construct more discriminative models that significantly improve the tracking performance.

Another related work is the distance metric learning that seeks an effective and discriminative metric space where both intra-class compactness and inter-class separability are maximized. Li *et al.* [33] proposed a metric-weighted linear representation of appearance to capture the interdependence of different feature dimensions and developed two online distance metric learning methods using proximity comparison

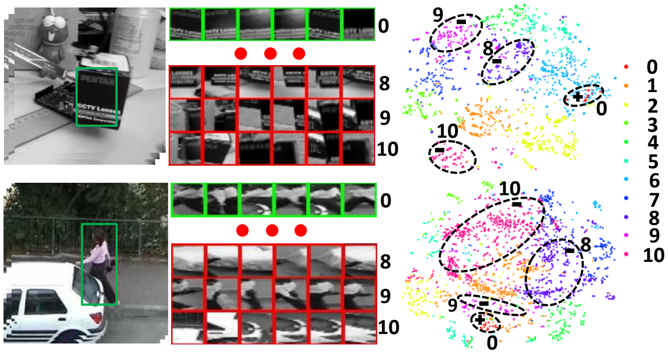


Fig. 2: Two instances of contextual clusters. Middle column: one foreground cluster (green) and ten contextual clusters (red). Each row corresponds to a separate cluster. Last column: the 2D layout of samples in principal components from the t-SNE dimension reduction [36] for visualization. 0: foreground. 1-10: color coded background clusters. As visible, there is a significant variance in the background samples, which may hinder the performance of a monolithic binary classifier.

information and structured output learning. Similarly, [34] observed that different visual metrics should be optimally learned for different candidate sets in the context of human reidentification problem, which is to match persons observed in non-overlapping camera views. This approach selects and reweights the training samples according to their visual similarities with the query sample and its candidate set. In contrast, our work does not handle the discriminative metric space. Instead, we explore the contextual information for the background samples via explicitly grouping them into clusters. We deploy a top-level SSVM to fuse the discriminative information from the individual clusters' SSVMs, which can be considered relevant to the metric learning concept.

III. Tracking with Multiple Clusters

The basic idea of tracking-by-detection is to establish object correspondence between consecutive frames using an object detector. Many recent state-of-the-art trackers are often based on this scheme [15], [35], [12], [16], resulting in improved accuracy and robustness of tracking performance. One reason is that an online updated classifier helps to address challenging situations such as appearance variations, partial occlusions, and background clutters in a single, unified manner.

Given an estimated object bounding box B_{n-1}^* in a previous frame $n-1$, the tracker proceeds to find a new object location B_n^* at the current frame n through a dynamically maintained and updated classification confidence function F as follows:

$$B_n^* = \operatorname*{arg\,max}_{B_n \in \mathcal{S}_n(B_{n-1}^*)} F_{n-1}(B_n), \qquad (1)$$

where $\mathcal{S}_n(B_{n-1}^*)$ denotes the set of candidates in frame n, sampled around the previous object location B_{n-1}^* within a search radius. For example, the search radius of 30 pixels was used in [35].

As mentioned above, to efficiently maintain and update a classification confidence function $F_{n-1} \rightarrow F_n$ is key to the success. In this regard, previous work (e.g. [12]) often use multiple-instance-learning to compose the positive and negative training samples (this can be also viewed as the online labeling task as in [35]) against label noise issue. However, many of these methods make the implicit assumption that the background (and the context) conform to a single, monolithic, possibly homogeneous class, which is rarely the case.

In contrast, our method does not assume the background samples have an identical distribution, or they belong to a single semantic class. We argue that, the appearance variance of the background can be better modeled by a committee of foreground-versus-contextual-cluster classifiers. Noticing in practice most tracking failures occur either when a background element such as background clutter distracts the tracker or when the tracker slightly drifts and starts accumulating error until a total breakdown, here we propose constructing fine-grained boundaries using multiple classifiers on contextual clusters.

A. Fine-Grained Classifiers

To better capture the latent data distributions of the negative samples (i.e. background clusters), which can be rather complex, we use unsupervised clustering with temporal continuity priors.

A simple way to perform this is to use k-means algorithm to label each negative sample as one of K clusters at every frame by initiating the iterations with the previously estimated clusters centers. Alternatively, Hough-forest based clustering [37] may be used. This method employs a random forest to cluster patches that have consistent appearance (and spatial displacement). Yet another solution is a graph mode-seeking method [7], which can automatically discover the distribution modes, i.e. dense subgraphs, of a graph characterized by a baseline kernel. In this work, we suggest the k-means algorithm mainly due to its computational simplicity.

An illustration of the clustering result is given in Figure 2. Here, the negative and positive sample set descriptors (480-dimensional intensity histogram features) are mapped down to a 2D space for visualization. We use a dimension reduction technique, t-distributed stochastic neighbor embedding (t-SNE) [36], which computes a mapping of distances while preserving the overall global structure. Notice that, each cluster of the background samples portrays *hard negative* patterns. Even though the background samples are collected from different frames, they exhibit patterns that can be clustered into a few consistent patterns, which will leverage the discriminative power of the corresponding classifiers.

We select SSVM as the foreground-versus-contextual cluster classifier, nonetheless our method can be extended to any object model easily. SSVM is shown to provide better object localization and tracking performance than other variants of SVM [38], [39].

Suppose the negative samples $\{\mathcal{B}_i \backslash B_i^* : i = 1, \ldots, n-1\}$ from $n-1$ previous frames are grouped into K contextual clusters $\{\{\mathcal{B}_i^k : i = 1, \ldots, n-1\} : k = 1, \ldots, K\}$, where $\mathcal{B}_i^k$ denotes the negative samples belonging to the k-th cluster, $\{B_i^* : i = 1, \ldots, n-1\}$ is the set of positives, and $\mathcal{B}_i$ is the set

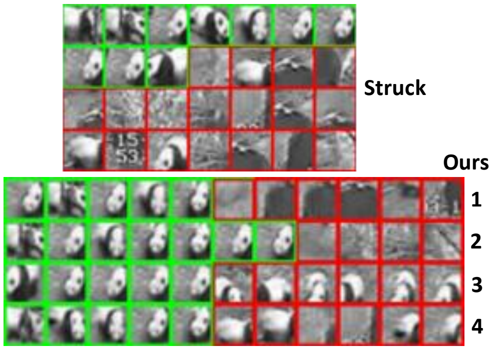


Fig. 3: Conventional single SSVM (top, from Struck [35]) versus the proposed multiple SSVMs of the contextual clusters (bottom, each row corresponds to one contextual SSVM). Green: positive support vectors. Red: negative support vectors. Notice that the burden of the classification task is reduced significantly for each contextual SSVM.

of all positive and negative samples at frame i. We separately train K classifiers to obtain confidence functions for each pair of the negative cluster $\{\mathcal{B}_i^k\}$ and the positive set $\{B_i^*\}$, which have the form of:

$$F_{n-1}^k(B_n) = \sum_{B_{i,j}^k \in \mathcal{V}_{n-1}^k} w_{i,j}^k \Phi(B_{i,j}^k, B_n) \quad k = 1, \dots, K, \quad (2)$$

where $\mathcal{V}_{n-1}^k$ is the support vector set of the k-th SSVM after the training process, and $w_{i,j}^k$ is a scalar weight associated with the support vector $B_{i,j}^k \in \mathcal{B}_i^k \cup B_i^*$ indexed by j from frame i. The kernel $\Phi(B_{i,j}^k, B_n)$ calculates the affinity between two feature vectors extracted from $B_{i,j}^k$ and B_n, respectively.

Here, both $\mathcal{V}_{n-1}^k$ and $w_{i,j}^k$ are learned using the online SSVM algorithm "Larank" [40], [41], which is shown to be an efficient SSVM solver [35]. As the image feature, we employ intensity histograms from a spatial pyramid [42] to represent the image patch in $\Phi(B_{i,j}^k, B_n)$ capturing the discriminative cues between the foreground and background patches. In the experiment section, we test different 2D kernels including linear, radial basis function (RBF) and intersection.

We ask the question whether a strong SSVM using RBF kernel with a large number of support vectors, most of which correspond to the previously estimated negative samples, would achieve the same performance. To our observations, simply inflating the number of support vectors does not generate a proportionally more accurate classifier since it either tends to overfit data or fails to model essential differences between the object and background samples. As shown in Table IV, a single very strong classifier results in only marginal improvement on the performance if any. In Figure 3, we give an example of the differences between the support vectors maintained by the tracker that uses a single strong SSVM [35] and by our proposed contextual SSVMs. It is apparent that the burden of the classification task is reduced for each contextual SSVM in our method, comparing to the single SSVM.

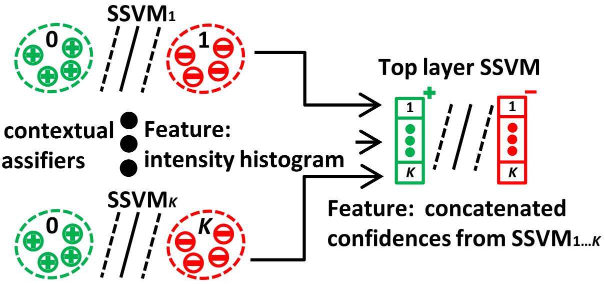

SSVM1
0
1
K contextual classifiers
Feature: intensity histogram
SSVMK
0
K
Top layer SSVM
+
−
1
K
1
K
Feature: concatenated confidences from SSVM1...K

Fig. 4: Hierarchy of classifiers in the proposed tracker. First layer: K separate SSVMs trained using the positive samples and the K contextual negative sample sets. Second layer: a single SSVM is trained to fuse the classification confidences. All SSVMs are updated online to adapt object appearance and background changes.

B. Confidence Combination

There are numerous strategies to combine multiple confidence functions including max or average pooling [43], voting, and multiple kernel learning. These, however, are not capable of learning an adaptive discriminative model for each video sequence.

Instead, we treat each confidence function as a feature generator, and use an additional top layer SSVM as illustrated in Figure 4 to learn the optimal combination of multiple confidence function results of K contextual clusters.

In this stage, the negative samples $\{\mathcal{B}_i \backslash B_i^* : i = 1, \dots, n-1\}$ and the positive samples $\{B_i^* : i = 1, \dots, n-1\}$ are used to train a top layer discriminant function $F_{n-1}(B_n)$ as:

$$F_{n-1}(B_n) = \sum_{B_{i,j} \in \mathcal{V}_{n-1}} w_{i,j} \Psi(B_{i,j}, B_n). \quad (3)$$

The difference between $F_{n-1}(B_n)$ and $F_{n-1}^k(B_n)$ is the design of the feature for kernel function $\Psi(B_{i,j}, B_n)$. This feature concatenates the classification confidences from the K SSVMs into a K-dimensional vector. Different choices of kernels are tested in the experimental section. The overall tracking algorithm is summarized in Algorithm 1.

C. Online Update with Temporally Consistent Clustering

In object tracking, the training data for the object model is given only in the first frame. The SSVM framework [40], [41] selects a triplet $\{i, B_{i,+}^k, B_{i,-}^k\}$ and optimizes their corresponding coefficients $w_{i,+}^k$ and $w_{i,-}^k$ using an SMO-style step [44]. The main step is to choose the negative support vector $B_{i,-}^k$ by

$$B_{i,-}^k = \arg\max_{B_i \in \mathcal{B}_i^k} \; L(B_i, B_i^*) + F_{n-1}^k(B_i) \quad (4)$$

where the loss function $L(B_i, B_i^*) = 1 - (B_i \cap B_i^*)/(B_i \cup B_i^*)$ defines on the bounding box overlap. Optimizing (4) corresponds to finding such a negative training sample that locates far from the positive one (high $L(B_i, B_i^*)$) yet presents close appearance (high $F_{n-1}^k(B_i)$). We use the C++ implementation from [35] for online optimization.

Algorithm 1. Two-layer SSVM based Tracker using Multiple Background Clusters

```
Tracking
Require: $K$ confidence functions $F^k_{n-1}$, the top layer discriminant function $F_{n-1}$, previous model $B^*_{n-1}$ and object location
  1. Generate $K$ confidence scores for each candidate in the search radius $\mathcal{S}_n(B^*_{n-1})$ in the current image: $F^k_{n-1}(B_n) = \sum_{B^k_{i,j}\in\mathcal{V}^k_{n-1}} w^k_{i,j}\Phi(B^k_{i,j}, B_n)$.
  2. Compute aggregated confidence score: $F_{n-1}(B_n) = \sum_{B_{i,j}\in\mathcal{V}_{n-1}} w_{i,j}\Psi(B_{i,j}, B_n)$.
Return: New location : $B^*_n = \arg\max_{B_n\in\mathcal{B}_n} F_{n-1}(B_n)$.

Update
Require: Support vector sets and the corresponding weights of $K$ contextual cluster SSVMs $\mathcal{V}^k_{n-1}$, and of the top layer SSVM $\mathcal{V}_{n-1}$, the new positive sample $B^*_n$ and negative samples $\mathcal{B}_n \backslash B^*_n$.
  1. Run $k$-means initialized with the previously estimated clusters centers to obtain the new contextual clusters: $\{\{\mathcal{B}^k_i : i = 1, \dots, n\} : k = 1, \dots, K\}$.
  2. Update the contextual SSVMs: $\mathcal{V}^k_n \leftarrow \mathcal{V}^k_{n-1}$ and the corresponding weights $w^k_{i,j}$ as in Section III-C.
  3. Update features for the top layer SSVM: $[F^1_n(B_i), \dots, F^K_n(B_i)], B_i \in \mathcal{B}_i, \forall i \in \{1, ..., n\}$, using the updated contextual SSVMs from 2.
  4. Train the top layer SSVM: $\mathcal{V}_n \leftarrow \mathcal{V}_{n-1}$ and the corresponding weights $w_{i,j}$ using online optimization with the features from step 3.
Return: Support vectors $\{\mathcal{V}^k_n \mid k = 1, \dots, K\}$, $\mathcal{V}_n$ and the corresponding weights.
```

To avoid independently re-clustering and re-optimizing over the K separate SSVMs at every frame, we benefit from the k-means initialization. First, we run the k-means multiple times to obtain a consistent clustering. At every new frame, we recycle the previous clusters' centers to initialize k-means clustering. Since only a portion of the previously clustered samples change after clustering, we keep the unchanged support vectors and avoid re-optimizing the SSVMs. To be specific, we use the *processold* step in [35] to add an extra number of negative support vectors, replacing those lost due to the re-clustering procedure if necessary.

Keeping all available training samples is not computationally and memory-wise efficient, thus we employ the budget management method used in [45]. This allows at most a fixed-number (100 in all experiments) of maintained support vectors. Once this number is exceeded, we remove the most insignificant support vectors that induce the smallest changes to the classification boundary.

IV. Experiments

Datasets Tested:

We evaluate our method on three recent benchmark datasets: OTB [15], TB50 [13] and VOT2014 [14]. These datasets provide a large number of sequences depicting a wide spectrum of challenging tracking scenarios.

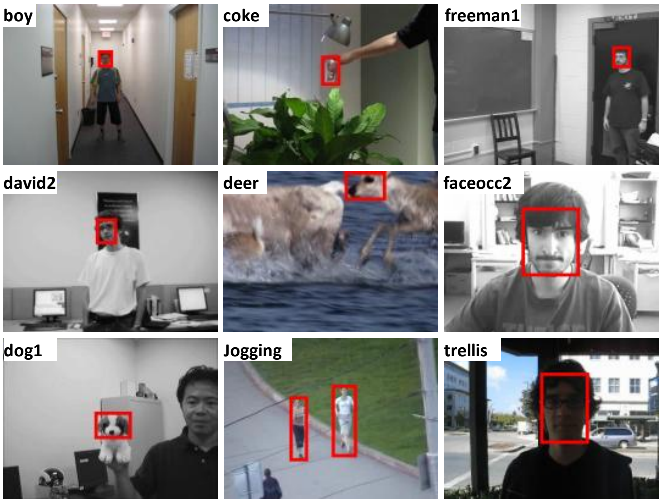


Fig. 5: Sample sequences from the OTB benchmark dataset [15] with ground truth object windows (red).

OTB contains 50 video sequences with fully bounding box annotations. The total number of frames is more than 29, 000, and for each sequence, the number varies from tens to thousands, e.g. *deer* (71 frames), *skiing* (81 frames), *dog*1 (1350 frames), *doll* (3872 frames), etc. A few sample sequences with ground truth annotations are shown in Figure 5.

In comparison to the OTB dataset, TB50 [13] contains more challenging sequences. Samples can be seen in Figure 7. Many of the TB50 sequences depict strong motion blur (e.g. *blurBody*), fast object motion (e.g. *dragonbaby*), and intermittent occlusions (e.g. *skating2*). As visible in Figure 6, there is a big performance gap for all trackers between OTB and TB50.

Both benchmarks additionally annotate each sequence globally with various visual attributes. Some common attributes available in the benchmarks are:

- Fast Motion - the motion of the ground truth is larger than t_m pixels ($t_m = 20$).
- Motion Blur - the target region is blurred due to the motion of target or camera.
- Deformation - non-rigid object deformation.
- Occlusion - the target is partially or fully occluded.

In the benchmarks, individual sequences are not per-frame annotated. For example, a sequence has the *occlusion* attribute if the target is occluded at any frame in the sequence. Although many factors could contribute to the performance, these attributes help us to diagnose the weaknesses and strengths in a more detailed way.

The sequences embodied in the VOT2014 benchmark are selected from widely used datasets in literature, including the Amsterdam Library of Ordinary Videos for tracking (ALOV++) [50] and OTB. It comprises a set of 25 sequences, which cover various real-life visual phenomena. The duration of these sequences are relatively short in order to keep the computational load of experimental evaluations reasonably

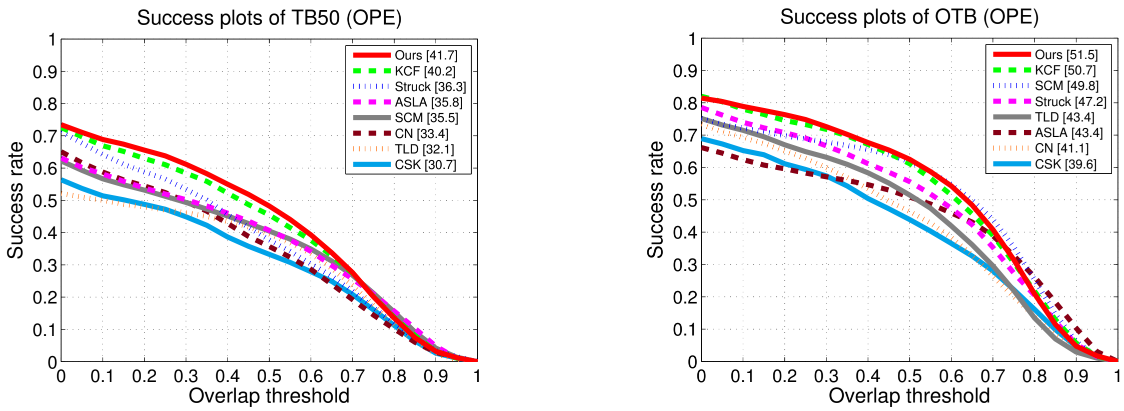


Fig. 6: *Success ratio* plots on the TB50 and OTB datasets. Trackers are ranked by the Area Under Curve (AUC) of the *success ratio* plots. As visible, our method (red) achieves the best performance on both datasets.

TABLE I: Area Under Curve (AUC) of *success ratio* plots and *precision scores* (at 20 pixels threshold) on TB50 and OTB benchmark datasets for the one-pass evaluation (OPE). fps: frames-per-second. Best in bold. Our performance on TB50 when we adapt to scale changes is even higher; **42.0/62.5**.

Datasets	Ours	Struck [35]	KCF [19]	SCM [46]	TLD [47]	CN [48]	ASLA [4]	CSK [49]
TB50 (50)	**41.7/61.2**	36.3/49.9	40.2/61.1	35.5/47.8	32.1/45.0	33.4/42.2	35.8/46.2	30.7/41.8
OTB (50)	**51.5**/72.5	47.2/65.3	50.7/**72.9**	49.8/64.8	43.4/60.1	41.1/55.3	43.4/60.1	39.6/54.1
fps	2.3	4.8	70.9	0.3	8.8	27.2	3.8	18.6

low. Unlike OTB and TB50, VOT2014 labels each frame in each sequence with five visual attributes. It also features a reinitialization evaluation scheme. After the tracker loses the target object during tracking, which is the case when the overlap measure with the ground truth becomes zero, the tracker is reinitialized five frames after the failure. This scheme measures the robustness of trackers by counting how many times they fail in a sequence.

Evaluation Metrics:

We use the metrics and the source code provided by these benchmarks. On OTB and TB50, the performance is evaluated using the *precision score* and *success ratio* metrics. The *precision score* calculates the rate of frames whose center location is within a certain threshold distance with the ground truth. Here, a commonly used threshold is 20 pixels as recommended by the benchmark protocol. This metric emphasizes how well a tracker is able to clasp the target. The *success ratio* calculates the same ratio based on bounding box overlap threshold $(B^* \cap B_{gt})/(B^* \cup B_{gt})$, where B^* and B_{gt} are the estimated and ground truth bounding boxes, respectively. This metric indicates how well a tracker adapts and covers the target. A typical value is 0.5 as used in object detection evaluation [55].

TABLE II: Robustness performance on VOT2014.

	Robustness Rank
Ours	**13.22**
DSST [51]	16.75
KCF [19]	17.95
SAMF [52]	17.81
MCT [14]	16.34
MUSTer [53]	18.49
MEEM [54]	16.42
Struck [35]	22.98

We employ the one-pass evaluation (OPE) that takes the ground truth at the first frame as the initialization bounding box then run trackers until the last frame.

For VOT2014, the benchmark provides a ranking based on the *robustness* performance measure. As mentioned above, the *robustness* measures how many times the tracker loses the target (failures). The ranking scheme considers the statistical significance of performance differences to ensure an objective comparison, e.g., trackers are equally ranked if there is only a negligible difference from a practical point of view. We also calculate the ranking result based on the *accuracy* metric, which measures how well the bounding box predicted by the tracker overlaps with the ground truth bounding box. We test all trackers 15 times on each sequence to obtain reliable statistics on performance measures.

Compared Methods:

OTB benchmark employed 29 recent and publicly available trackers and TB50 employed 31 trackers. For clarity in the performance graphs, we compare our method against the top ranked trackers in these datasets including Struck [35], SCM [46], TLD [47], ASLA [4] and CSK [49]. We additionally compare with two other recent trackers: KCF [19] and CN [48], which report strong performance. Struck [35] uses a similar SSVM framework. Unlike our method, it treats tracking as a single, binary foreground-versus-background classification problem thus can be considered as the baseline tracker. KCF [19] tackles the undersampling issue with circulant matrices. Similar to Struck, KCF tracker considers a single background model.

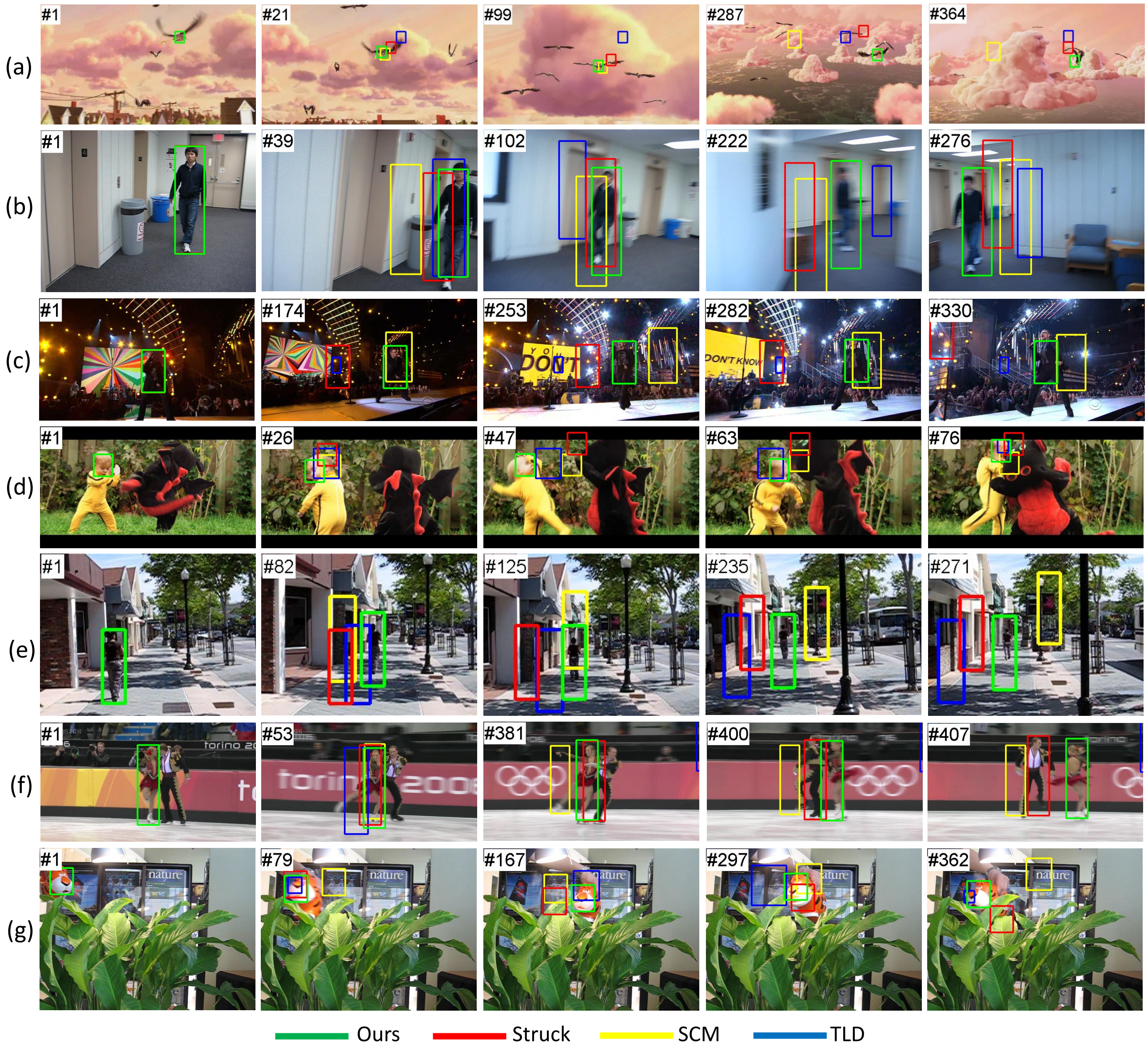


Fig. 7: Qualitative comparisons with the state-of-the-art trackers on videos from TB50. (a) Bird1; (b) BlurBody; (c) Singer2; (d) DragonBaby; (e) Human9; (f) Skating2; (g) Tiger2. Our method attains robust tracking performance in challenging scenarios including fast motion, motion blur, deformation, and occlusion. Notice that, the object window size in each video is fixed.

VOT2014 challenge collected 38 trackers for evaluation, including KCF and Struck. For readability of the performance graphs, we compare with the top ranked trackers on VOT2014 with two additional methods, MEEM [54] and MUSTer [53].

We use the publicly available code and default settings from the original authors for a fair comparison.

Benchmark Results:

The tracking results for the benchmark datasets are presented in Table I, Figure 6, Table II and Figure 8.

As shown, our method outperforms all other trackers including more recent approaches CN and KCF on both TB50 and OTB in both the precision score and the Area Under Curve (AUC) of the success plot. On VOT2014, our method achieves the best robustness rank among all state-of-the-art. It exhibits consistent performance for all three benchmarks as well.

Our method of using multiple backgrounds also significantly improves its baseline tracker (Struck). On the OTB dataset, our improvement is significant; 4.3% for the AUC and 7.2% for the precision score. On the more challenging TB50 dataset, we achieve even a greater improvement; 5.4% for the AUC and 11.3% for the precision score. On VOT2014, our method boosts the robustness rank from 22.98 to the best score 13.22. These results demonstrate that our multiple-contextual-clusters method remarkably benefits discriminative classification schemes for tracking.

Sample tracking results of our method and the top performing state-of-the-art trackers are given in Figure 7 for qualitative analysis. As visible, our method tracks the target objects accurately over many various challenging scenarios, where all others fails (e.g., Struck,, SCM, TLD, etc.).

Attributes (TB50)	Ours	Struck [35]	KCF [19]	SCM [46]	TLD [47]	CN [48]	ASLA [4]	CSK [49]
FM (25)	**41.6**/**59.5**	34.4/42.5	39.0/54.0	25.2/29.6	35.6/46.5	30.9/35.2	25.0/26.0	26.4/33.7
MB (19)	**42.4**/**59.2**	30.9/35.5	40.6/56.4	21.7/25.1	39.3/49.7	31.1/36.0	23.3/25.5	29.8/36.4
DEF (23)	**43.6**/**65.0**	32.5/41.5	39.8/58.2	28.5/40.3	24.8/33.4	32.1/35.9	34.7/46.9	25.8/33.4
IPR (29)	**41.4**/58.0	34.3/45.2	38.7/**58.7**	34.5/46.2	33.1/45.8	36.4/48.5	33.9/43.9	29.8/40.6
OPR (32)	**40.2**/**60.3**	35.3/49.2	39.5/59.8	35.5/49.1	29.0/41.3	32.6/42.4	38.0/49.2	26.2/36.4
OV (11)	**42.4**/**68.5**	33.9/46.1	32.8/44.1	27.9/35.8	30.8/41.6	30.6/35.5	31.0/38.3	21.3/25.6
OCC (29)	**40.5**/**62.0**	35.6/49.8	39.5/60.4	34.8/48.0	27.4/39.5	32.0/40.7	36.7/48.5	26.5/37.3
BC(20)	39.0/55.2	36.5/47.6	**41.7**/**62.3**	36.2/46.6	29.5/39.9	35.6/43.7	39.8/50.0	34.3/46.7
SV (38)	35.9/54.7	34.0/47.5	35.2/**56.5**	**37.0**/49.7	30.0/42.4	32.4/35.9	35.8/46.3	26.7/36.6

TABLE III: Area Under Curve (AUC) of *success ratio* plots and *precision scores* (at 20 pixels threshold) on TB50 dataset attributes. FM: fast motion, MB: motion blur, DEF: deformation, IPR: in-plane rotation, OPR: out-of-plane rotation, OV: Out-of-view, BC: background clutters, SV: scale variation. Best results are shown in bold.

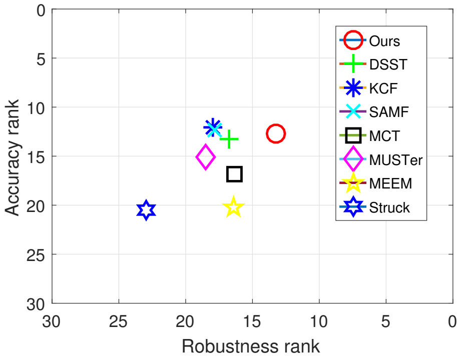


Fig. 8: Accuracy-robustness ranking plots of our method and top ranked methods on VOT2014. Our tracker provides the best trade-off between accuracy and robustness

To demonstrate that simply increasing the complexity of a single classifier is not an effective model for tracking and thus cannot achieve a better performance as our method, we evaluated the performance of Struck [35] with increased number of support vectors. The results are given in Table IV where Struck$_{500}$ denotes the singe SSVM based tracker using a maximum of 500 support vectors. The original Struck uses 100 support vectors. It is apparent that insignificant improvement is obtained by increasing the number of support vectors albeit considerable computational expense.

Performance on Attribute Categories:

To obtain a better understanding, we evaluated the performance of our method on the attribute categories of TB50. Comparative results are given in Table III, Figure 9 and 10.

For most attributes, such as *motion blur*, *fast motion* and *deformation*, our method achieves superior performance. For *motion blur* and *fast motion*, the performance improvement comes from the fact that our method elegantly instantiates specific trackers for the contextual cluster of hard negative samples for the shifted version on the object window, which provides enhanced localization accuracy. For *deformation*, our method allows efficiently distributing the burden of modeling the foreground object variations over multiple classifiers, which would be difficult for a single SSVM to distinguish. For *background clutter* and *scale change*, we have still significantly better results than the base tracker, i.e. Struck.

TABLE IV: Struck [35] performance on TB50 for different maximum number of support vectors.

	Ours	Struck$_{100}$	Struck$_{200}$	Struck$_{500}$
AUC/PS	**41.7**/**61.2**	36.3/49.9	35.9/50.6	36.4/50.9
fps	2.3	4.8	4.3	3.7

Implementation Details and Variants:

Our method uses the intersection kernel for confidence function of the contextual cluster SSVMs, the linear kernel for the top layer discriminant classifier, and intensity histogram as low-level features.

We employ the motion model that applies a 2D translation $\{(u, v)|u^2 + v^2 < r^2\}$ for simplicity. During tracking we apply a search radius $r = 30$ pixels and during updating the classifier we take a larger radius $r = 60$ to incorporate possible nearby hard negatives in the negative samples and to ensure robustness. We sample candidate object locations on a polar grid (5 radial and 16 angular divisions, giving 81 locations).

The classification models for both the contextual cluster SSVMs and the top layer discriminant SSVM are online updated every 5 frames to trade off between computational efficiency and robustness. The algorithm parameters involved in online updating SSVM using "LaRank" [41] are set similar to [35] for a fair comparison.

As feature, we operate with concatenated 16-bin intensity histograms from a spatial pyramid of 4 levels. At each pyramid level l, the underlying patch is divided into $l \times l$ cells, resulting in a $D = 480$ dimensional feature vector $[h_B^1, ..., h_B^D]$. We also tested the variants using different image features such as Haar wavelets and raw image patch. For the features we analyzed:

- Haar feature - 6 different types of Haar-like feature arranged on a grid at 2 scales on a 4×4 grid, resulting in 192-D features, with each feature normalized to give a value in the range $[-1, 1]$.
- Raw patch - Raw pixel features obtained by scaling a patch to 16×16 pixels and taking the greyscale value (in the range $[0, 1]$). This gives a 256-D feature vector.

TABLE V: Different low-level features. Results on TB50.

	Histogram	Haar	Raw Intensity
AUC/PS	**41.7**/**61.2**	39.8/56.1	40.1/58.6
fps	2.3	3.5	3.1

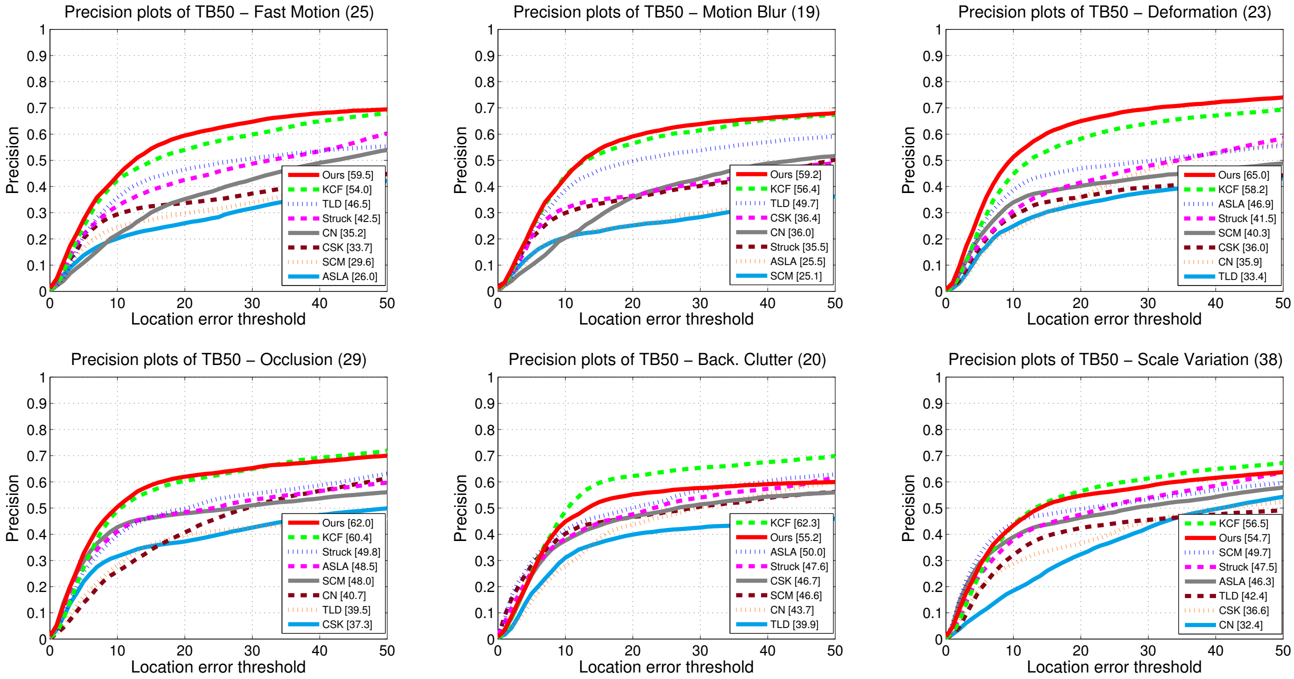


Fig. 9: *Precision score* plots on various attribute categories of the TB50 dataset. Trackers are ranked by their precision score at 20 pixels threshold.

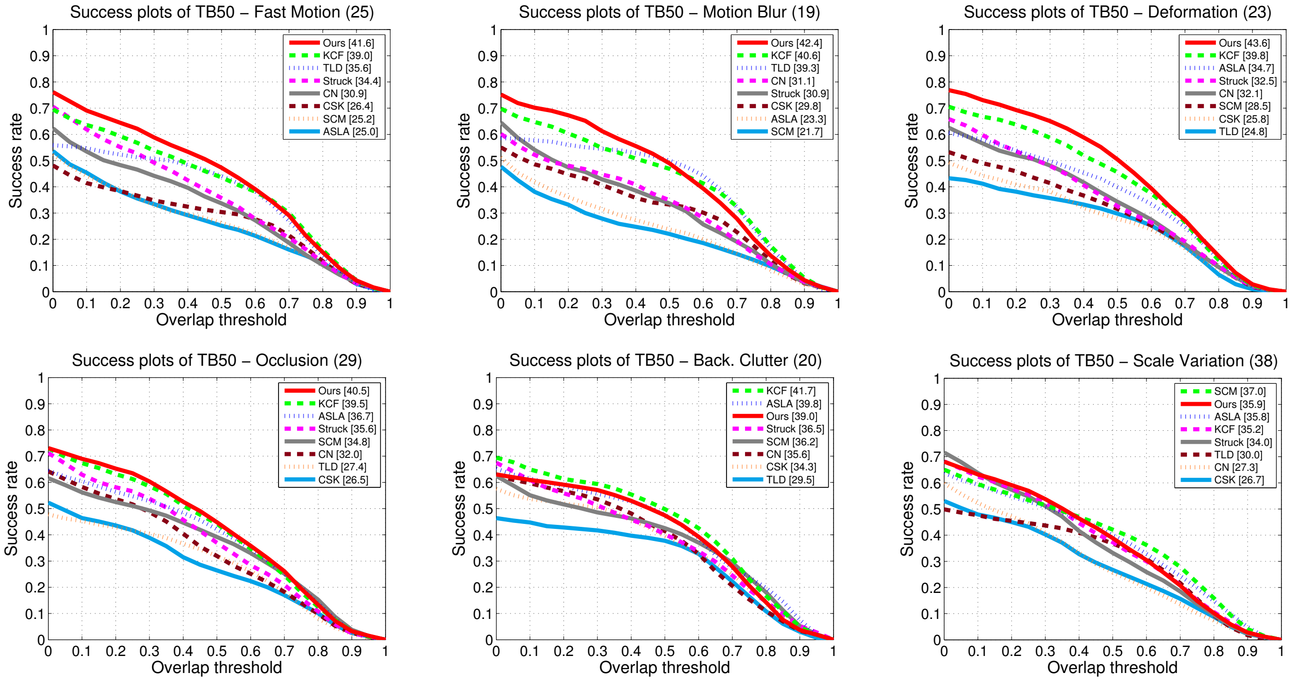


Fig. 10: *Success ratio* plots on six attribute categories of the TB50 dataset. Trackers are ranked by their AUC scores. Ours method has achieved consistently superior performance in various categories.

The comparison of features are available in Table V. Remarkably, the raw intensity feature performed better than the Haar feature. One explanation is that the Haar feature is not sensitive enough to the discriminative yet fine-grained appearance details.

To further evaluate our method, we examine the effectiveness of the top layer SSVM by replacing it with commonly used pooling methods. As discussed in in Section III-B, the incorporated top layer SSVM is for combining the confidence function results from the contextual cluster SSVMs. As an

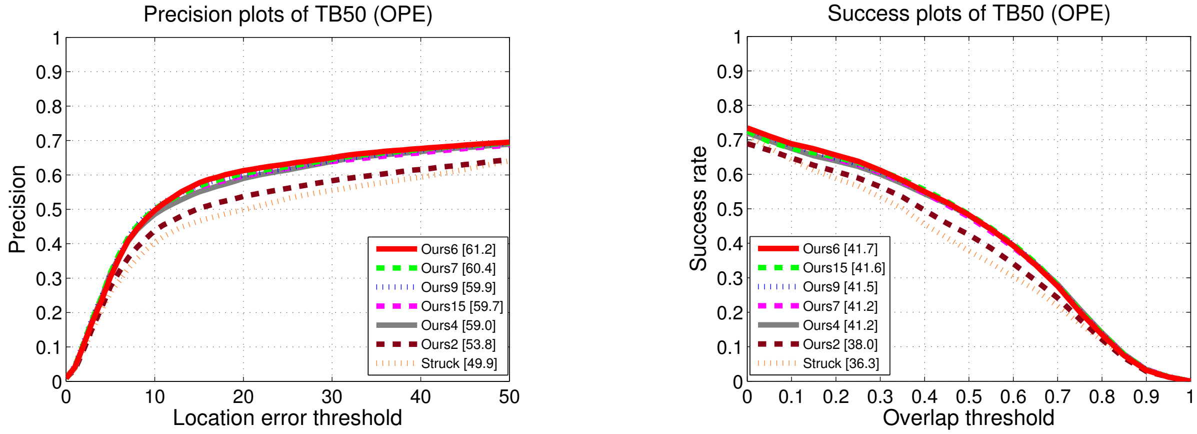


Fig. 11: *Success ratio* and *precision score* plots of our method with different number of clusters. All our variants are better than Struck.

alternative, we test three different pooling methods to fuse the confidence scores: mean pooling, median pooling and maximum pooling. The results are shown in Table VI. As we can see from the results, all pooling methods cause inferior performance compared to ours. This is expected as the incorporated top layer SSVM learns in an online fashion to trust which contextual cluster classifier instead of blindly and heuristically choosing one.

TABLE VI: Use of different pooling schemes instead of the top layer discriminant SSVM. Results on TB50.

	Ours	mean	median	max
AUC/PS	**41.7/61.2**	39.8/57.3	39.3/58.2	40.6/59.3
fps	2.3	3.1	3.0	3.1

We also analyzed the alternative kernel combinations for the contextual SSVMs and the top layer SSVM. The joint kernel function $\Phi(B^k_{i,j}, B_n)$ (2) is implemented using the intersection kernel:

$$\Phi(B^k_{i,j}, B_n) = \frac{1}{D}\sum_{d=1}^{D} \min(h^d_{B^k_{i,j}}, h^d_{B_n}).$$

We use the linear kernel for the top layer discriminant function $\Psi(B_{i,j}, B_n)$ (3), which computes the inner products. Results are shown in Table VII. In this experiment, we set $\sigma = 0.1$ for the Gaussian kernel. We observed that the linear kernel generates inferior results when used in the contextual SSVMs, however it gives the best accuracy when used in the top layer SSVM. This is possibly due to the fact that the feature complexity is significantly different between these two layers.

TABLE VII: Different kernels. Results on TB50.

Contextual SSVMs	Linear 38.1/52.3	Gaussian 41.1/60.6	Intersection **41.7/61.2**
Top Layer SSVM	Linear **41.7/61.2**	Gaussian 40.3/58.7	Intersection 40.7/59.1

For k-means, the cluster number is set to $K = 6$ for all experiments. We also tested variants using different cluster numbers. The results can be seen in Figure 11. As visible in the graphs, our method is robust against the cluster number changes, and always better than using a single cluster. This validates the use of multiple clusters, and multiple classifiers, for the background samples.

We additionally investigated combining the spatial coordinates of samples with the visual features to enforce spatial consistency of samples within each cluster. We observed that this does not improve the performance. Besides, a heuristic imposition of spatial closeness of samples within the clusters escalates maintenance issues of clusters, in particular when the object motion causes the background to change.

Size Adaptation:

Our method uses a simple fixed object bounding box representation through the tracking process as Struck [35] and KCF [19]. Yet, it is straightforward to extend our method to adapt scale and aspect ratio changes by modifying the motion model from the 2D translation $\{(u, v)|u^2 + v^2 < r^2, r = 30\}$ to a 3D or 4D motion models (with scale and aspect ratio changes: step 0.1, range $[0.8, 1.2]$). The results are reported in Table VIII and sample detections are depicted in Figure 12.

TABLE VIII: Adaptation of size change. Results on TB50.

	Ours (fixed)	Scale	Scale+As.Ra.
AUC/PS	41.7/61.2	**42.0/62.5**	41.5/60.2
fps	2.3	1.1	0.4

As visible, scale adaption further improves the AUC/PS on TB50. Yet, this increases the computational cost. By adapting scale, the performance may potentially improve for the *scale variation* category. This can be validated from Table III, where SCM (size adapted) gives better scores for the *scale variation* category. However, for attributes such as *occlusion* and *deformation*, trackers with fixed object size tend to perform more robustly.

Possible Failure Cases:

As we can see from Table III, our method performs superior in most benchmark attributes, however it is among the second

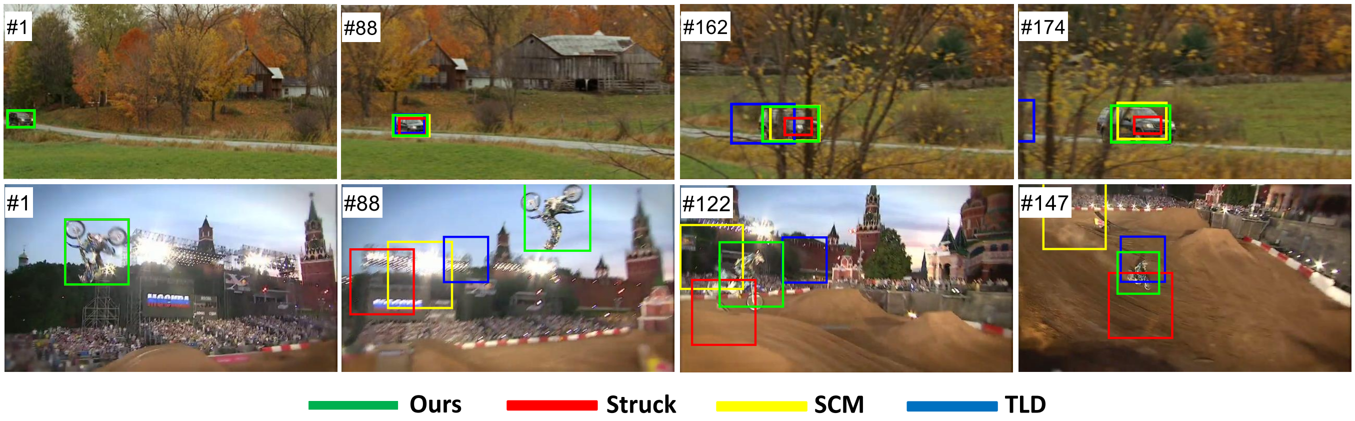


Fig. 12: Size change adaptation: sample results of our method and the state-of-the-art trackers on videos from TB50. Top row: CarScale; Bottom row: MotorRolling.

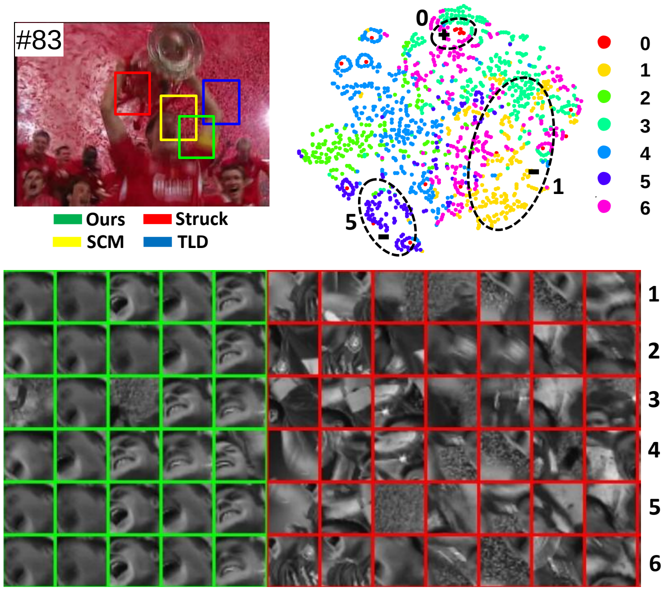


Fig. 13: A failure example from sequence 'Soccer' (TB50). Upper right: 2D layout (t-SNE) of k-means results of training samples. 0: foreground. 1-6: color coded background clusters. Bottom: each row corresponds to one contextual SSVM. Green: positive support vectors. Red: negative support vectors.

best trackers for the *scale variation* and *background clutter* after KCF. One reason for this is that we employed fixed bounding box sizes, which may have limited its capacity to acquire correct foreground models when the target object undergoes drastic scale changes. For the *background clutter*, the reason could be that there is no apparent distribution of multiple clusters exhibited as shown in Figure 13. In this case, k-means may fail to extract effective contextual clusters as illustrated in the 2D layout of the clusters and support vectors of the contextual SSVMs. Notice that, k-means has a random nature that may lead to this.

Nevertheless, our method of incorporating multiple contextual background clusters is always better than Struck for all attributes. This corroborates the robustness of our hierarchical SSVM structure regardless of unstable clustering results of k-means. We argue that all clusters are subsets of the background samples, and even potentially irregular clusters contribute to foreground-background classification task, thus their responses do not deteriorate the second layer's prediction capacity.

Computational Complexity:

Our method is implemented in C++ and experiments are carried out on an Intel Core i7 3.40GHz PC with 4GB memory. Computational time is reported in Table I. The speed of our method is 2.3 fps on average without any optimization. The overall computational cost is comparable to existing methods. In addition, it is not increased significantly in comparison to the single SSVM (e.g. [35]) despite we use additional SSVMs. The reason is that the most time-consuming part in our method is in the optimizing (4) stage, i.e. exhaustively searching over the negative sample space to find a negative support vector as shown in Section III-C. In our method, for each foreground-versus-contextual cluster SSVM, this search space is greatly reduced.

V. Conclusions

We presented a tracking method that tackles the object detection task by designating multiple classifiers where each targets discriminating a different background cluster from object samples, and combining their responses into a top layer identifying to which pattern of classifier responses indicate object. This significantly reduces the burden on the classifier, allows learning of fine-grained yet important decision boundaries, and lends itself to efficient and accurate adaption to object and background changes.

By explicitly grouping the negative samples into multiple clusters, building multiple foreground-versus-cluster SSVM classifiers, and employing another single SSVM to learn the best combination of the confidences generated from the respective contextual classifiers, the proposed method achieves superior discriminative power as verified on standard benchmark datasets.

References

[1] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1-3, pp. 125–141, 2008. 1

[2] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 1436–1443. 1

[3] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 1305–1312. 1

[4] X. Jia, H. Lu, and M. H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1822–1829. 1, 6, 8

[5] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang, "Incremental learning of 3D-DCT compact representations for robust visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 863–881, 2013. 1

[6] S. Avidan, "Support vector tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2001, pp. 1064–1072. 1, 2

[7] X. Li, A. Dick, H. Wang, C. Shen, and A. van den Hengel, "Graph mode-based contextual kernels for robust SVM tracking," *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 1156–1163, 2011. 1, 2, 3

[8] F. Yang, H. Lu, and M. H. Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 242–254, 2014. 1

[9] G. Zhu, F. Porikli, Y. Ming, and H. Li, "Lie-Struck: Affine tracking on Lie groups using structured SVM," in *Proc. IEEE Winter Conf. on Appli. of Comput. Vis. (WACV)*, 2015. 1

[10] G. Zhu, F. Porikli, and H. Li, "Tracking randomly moving objects on edge box proposals," *CoRR*, 2015. 1

[11] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: parallel robust online simple tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 723–730. 1

[12] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2011. 1, 3

[13] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015. 2, 5

[14] M. Kristan et al., "The visual object tracking VOT2014 challenge results," in *Proc. Euro. Conf. Comput. Vis. Workshops (ECCVW)*, 2014. 2, 5, 6

[15] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013, pp. 2411–2418. 2, 3, 5

[16] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, 2013. 2, 3

[17] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014. 2

[18] M. Tian, W. Zhang, and F. Liu, "On-line ensemble SVM for robust object tracking," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2007, pp. 355–364. 2

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015. 2, 6, 8, 10

[20] X. Li, A. Dick, C. Shen, Z. Zhang, A. van den Hengel, and H. Wang, "Visual tracking with spatio-temporal Dempster-Shafer information fusion," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3028–3040, 2013. 2

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, 2015. 2

[22] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations by convolutional neural networks for visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014. 2

[23] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015. 2

[24] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," *CoRR*, 2015. 2

[25] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1195–1209, 2009. 2

[26] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning where the object might be," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 1285–1292. 2

[27] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 1177–1184. 2

[28] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 2113–2120. 2

[29] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 854–869, 2007. 2

[30] T.-K. Kim and R. Cipolla, "MCBoost: Multiple classifier boosting for perceptual co-clustering of images and visual features." in *Neural Information Processing Systems (NIPS)*, 2008, pp. 841–856. 2

[31] M. Godec, S. Sternig, P. M. Roth, and H. Bischof, "Context-driven clustering by multi-class classification in an active learning framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2010, pp. 19–24. 2

[32] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof, "Online multi-class LPBoost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 3570–3577. 2

[33] X. Li, C. Shen, A. Dick, Z. Zhang, and Y. Zhuang, "Online metric-weighted linear representations for robust visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016. 2

[34] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds., 2012. 3

[35] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 263–270. 3, 4, 5, 6, 8, 10, 11

[36] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008. 3

[37] J. Gall, A. Yao, N. Razavi, L. V. Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2188–2202, 2011. 3

[38] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *Proc. Euro. Conf. Comput. Vis. (ECCV)*, 2008, pp. 2–15. 3

[39] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, 2005. 3

[40] A. Bordes, L. Bottou, P. Gallinari, and J. Weston, "Solving multiclass support vector machines with LaRank," in *Proc. IEEE Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 89–96. 4

[41] A. Bordes, N. Usunier, and L. Bottou, "Sequence labelling SVMs trained in one pass." in *Euro. Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2008, pp. 146–161. 4, 8

[42] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2006, pp. 2169–2178. 4

[43] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2011, pp. 1–12. 4

[44] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998. 4

[45] Z. Wang, K. Crammer, and S. Vucetic, "Multi-class Pegasos on a budget," in *Proc. IEEE Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 1143–1150. 5

[46] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1838–1845. 6, 8

[47] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 49–56. 6, 8

[48] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1090–1097. 6, 8

[49] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Euro. Conf. Comput. Vis. (ECCV)*, 2012, pp. 702–715. 6, 8

[50] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468. 5

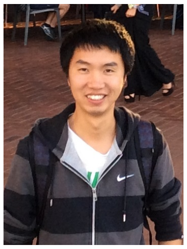
[51] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014. 6

[52] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Euro. Conf. Comput. Vis. Workshops (ECCVW)*, 2014. 6

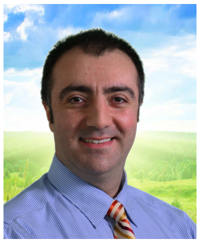
[53] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (must," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. 6, 7

[54] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Euro. Conf. Comput. Vis. (ECCV)*, 2014. 6, 7

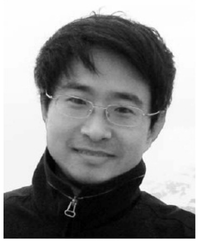
[55] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015. 6



Gao Zhu is currently pursuing his Ph.D. degree with the Computer Vision and Robotics Group, College of Engineering and Computer Science, Australian National University (ANU), Canberra, Australia. He has been working on object detection, object segmentation and visual object tracking problems. He received his M.Sc. degree from the University of Science and Technology of China, Hefei, China, in 2012. He serves as a reviewer for WACV'15, ICCV'15, WACV'16, CVPR'16, ACCV'16 and ECCV'16.



Fatih Porikli is an IEEE Fellow and a Professor with the Research School of Engineering, Australian National University, Canberra, Australia. He is also acting as the Leader of the Computer Vision Group at NICTA, Australia. He received his Ph.D. degree from NYU. Previously he served as a Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge, USA. He has contributed broadly to object detection, motion estimation, tracking, image-based representations, and video analytics. He is the coeditor of two books on Video Analytics for Business Intelligence and Handbook on Background Modeling and Foreground Detection for Video Surveillance. He is an Associate Editor of five journals. His publications won four Best Paper Awards and he has received the R&D100 Award in the Scientist of the Year category in 2006. He served as the General and Program Chair of numerous IEEE conferences in the past. He has 66 granted patents.



Hongdong Li is with the Australian National University, Canberra, Australia. His research interests include 3D computer vision, image analysis and pattern recognition, and mathematical optimization. He is the Chief Investigator and a Deputy Theme Leader of the ARC Centre for Robotic Vision, and was a member of the Bionic Vision Australia. Prior to 2010, he was a Senior Researcher with NICTA, and a Fellow of the Research School of Information Sciences and Engineering (RSISE) at ANU. He was a recipient and co-author of the CVPR Best Paper Award in 2012, the ICIP Best Student Paper Award in 2014, the ICPR Best Student Paper Award in 2010, DSTO Fundamental Paper Award and the DICTA CSiRA Best Theory Paper Award in 2013. He serves on various program committees (Area Chair) for recent ICCV, CVPR, ECCV and BMVC.