Robust Superpixel Tracking

Fan Yang, *Student Member, IEEE*, Huchuan Lu, *Senior Member, IEEE*,
and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— While numerous algorithms have been proposed for object tracking with demonstrated success, it remains a challenging problem for a tracker to handle large appearance change due to factors such as scale, motion, shape deformation, and occlusion. One of the main reasons is the lack of effective image representation schemes to account for appearance variation. Most of the trackers use high-level appearance structure or low-level cues for representing and matching target objects. In this paper, we propose a tracking method from the perspective of midlevel vision with structural information captured in superpixels. We present a discriminative appearance model based on superpixels, thereby facilitating a tracker to distinguish the target and the background with midlevel cues. The tracking task is then formulated by computing a target-background confidence map, and obtaining the best candidate by maximum *a posterior* estimate. Experimental results demonstrate that our tracker is able to handle heavy occlusion and recover from drifts. In conjunction with online update, the proposed algorithm is shown to perform favorably against existing methods for object tracking. Furthermore, the proposed algorithm facilitates foreground and background segmentation during tracking.**

***Index Terms*— Visual tracking, superpixel, appearance model, midlevel visual cues.**

I. Introduction

THE recent years have witnessed significant advances in visual tracking with the development of efficient algorithms and fruitful applications. Examples abound, ranging from algorithms that resort to low-level visual cues to high-level structural information with adaptive models to account for appearance variation as a result of object motion [1]–[14]. While low-level cues are effective for feature tracking and scene analysis, they are less effective in the context of object tracking [15]–[17]. On the other hand, numerous works have demonstrated that adaptive appearance models play a key role in achieving robust object tracking [1]–[3], [5], [12], [18], [19]. In recent years, mid-level visual cues have been applied to numerous vision problems including object segmentation [20]–[22], recognition [20], [23], pose estimation [24]. Nevertheless, much less attention is paid to exploit mid-level visual cues for visual tracking in complex scenes. To account for large appearance variation, it is of great interest to develop adaptive appearance model based on mid-level visual cues.

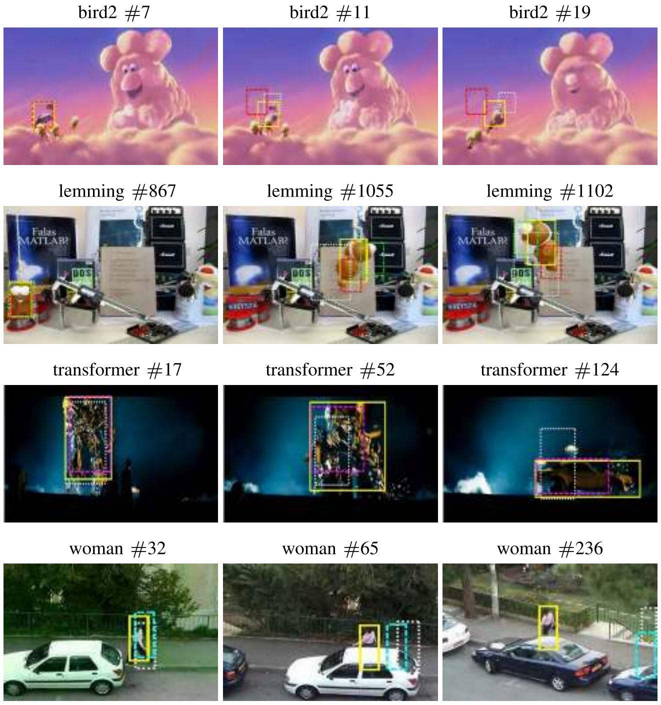


Fig. 1. Four challenges encountered in tracking. The results by our tracker, IVT [3], VTD [12], PROST [18], Frag [5], and PDAT [8] methods are represented by yellow, red, white, green, cyan, and magenta rectangles. Existing trackers are not able to effectively handle heavy occlusion, large variation of pose and scale, and non-rigid deformation, while our tracker gives more robust results.

In this paper,[1] we exploit effective and efficient mid-level visual cues for object tracking with superpixels (see Fig. 1). We present a discriminative appearance model based on superpixels, thereby facilitating a tracker to distinguish the target and the background with mid-level cues. The tracking task is then formulated by computing a target-background confidence map, and obtaining the best candidate by maximum a posterior estimate. During the training stage, the segmented superpixels are grouped for constructing a discriminative appearance

Manuscript received May 12, 2013; revised September 16, 2013 and November 14, 2013; accepted November 30, 2013. Date of publication January 21, 2014; date of current version March 3, 2014. This work was supported by joint Foundation of China Education Ministry and China Mobile Communication Corporation under Grant MCM20122071. The work of M.-H. Yang was supported in part by NSF CAREER Grant 1149783 and NSF IIS Grant 1152576. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark H.-Y. Liao.

F. Yang and H. Lu are with the School of Information and Communication Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: fyang@umiacs.umd.edu; lhchuan@dlut.edu.cn).

M.-H. Yang is with the School of Engineering, University of California, Merced, CA 95344 USA (e-mail: mhyang@ucmerced.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2300823

[1] Preliminary results of this work were presented in [25].

model to distinguish foreground objects from cluttered backgrounds. In the test phase, a confidence map at superpixel level is computed using the appearance model to obtain the most likely target location with maximum a posteriori (MAP) estimates. The appearance model is constantly updated to account for variation caused by change in both the target and the background. We also include a mechanism to detect and handle occlusion in the proposed tracking algorithm for adaptively updating the appearance model without introducing noise. Experimental results on various sequences show that the proposed algorithm performs favorably against existing state-of-the-art methods. In particular, our algorithm is able to track objects undergoing large non-rigid motion, rapid movement, large variation of pose and scale, heavy occlusion and drifts. As a by-product, we show that our algorithm is able to carry out foreground/background segmentation during tracking.

II. Related Work and Problem Context

In this section, we discuss the related online tracking algorithms and put our work in proper context. Online appearance models have been developed and applied to object tracking in recent years. In [1], an adaptive mixture model is proposed to deal with appearance change where the responses of wavelet filters are modeled with three components. While this method is able to track objects with illumination change and brief occlusion, this generative model considers pixels within the target region independently and does not exploit discriminative classifiers for separating foreground objects and the background. A kernel-based tracking algorithm that selects discriminative features to separate the foreground objects, modeled as a blob, and the background has shown to be effective in object tracking [2]. In [3], an incremental visual tracker (IVT) with adaptive appearance model that aims to account for appearance variation of rigid or limited deformable motion is presented. Although it has been shown to perform well when target objects undergo lighting and pose variation, this method is less effective in handling heavy occlusion or non-rigid distortion as a result of the adopted holistic appearance model. The ensemble tracker [26] formulates the task as a pixel-based binary classification problem. Although this method is able to differentiate between target and background, the pixel-based representation is rather limited and thereby constrains its ability to handle heavy occlusion and clutter.

Numerous algorithms have being proposed using local and multiple representation schemes to account for appearance change and occlusion. The fragment-based (Frag) tracker [5] aims to solve partial occlusion with a representation based on histograms of local patches. The tracking task is carried out by combining votes of matching local patches using a template. Nevertheless, the template is not updated and thereby it is not expected to handle appearance change due to large variation in scale and shape deformation. In [7], an algorithm extends multiple instance learning to an online setting for object tracking. While it is able to reduce drifts, this method is not able to handle large non-rigid shape deformation. The ℓ_1 tracker [11] first applies sparse representation to visual tracking with designed trivial templates to handle occlusions. Based on the templates, the ℓ_1 tracker poses the tracking problem as finding the image region with minimal reconstruction error using ℓ_1 minimization. With a generative formulation, the ℓ_1 tracker does not exploit the appearance information from the background and thus it is ineffective in handling heavy occlusion. The visual tracking decomposition (VTD) approach effectively extends the conventional particle filter framework with multiple motion and observation models to account for appearance variation caused by change of pose, lighting and scale as well as partial occlusion [12]. Nevertheless, as a result of the adopted generative representation scheme, this tracker is not equipped to distinguish target and background patches. Consequently, background pixels within a rectangular template are inevitably considered as parts of foreground object, thereby introducing significant amount of noise in updating the appearance model.

Object tracking can also be posed as a detection problem with local search. The PROST method [18] extends the tracking-by-detection framework with multiple modules for reducing drifts. Although this tracker is able to handle certain drifts and shape deformation, it is not clear how this method can be extended to handle targets undergoing non-rigid motion or large pose variation. In [13], a binary classifier is learned by using the structure of unlabeled data with positive and negative constraints. This classifier is used as an object detector for object tracking with online update. As a tracking-by-detection approach, this algorithm can re-detect the object when it disappears, so it is able to handle occlusion to some extent but not able to deal with pose and scale change well. More recently, a structured output tracking (Struck) method [14] is proposed by adopting kernelized structured output support vector machine to avoid the labeling ambiguity when updating the classifier during tracking. With simple low-level features, this method is less effective in handling scale change and occlusion.

Compared with high-level appearance models and low-level features, mid-level visual cues have been shown as effective representations containing sufficient information of image structure. In particular, superpixels have been applied to image segmentation and object recognition with demonstrated success [20], [23], [24], [27], [28]. These methods are able to segment images into numerous superpixels with evident boundary information of object parts from which effective representations can be constructed. In [22], a tracking method based on superpixels is proposed, which regards tracking task as a figure/ground segmentation across frames. However, as it processes every entire frame individually with Delaunay triangularization and conditional random filed for region matching, the computational complexity is rather high. Furthermore, it is not designed to handle complex scenes including heavy occlusion and cluttered background as well as large lighting change. Similarly, a non-parametric formulation is presented to model foreground and background classes for localizing and segmenting objects [21] in image sequences. We note that these methods are mainly developed for figure-ground separation and unlikely to perform well in cluttered scenes with large illumination change.

In addition to superpixels, several segmentation-based tracking algorithms have been proposed. In [29], a probabilistic tracking method using a bag-of-pixels representation and rigid transformation is proposed. Segmentation is used to obtain the object shape and estimate the probabilistic distributions of the foreground and background regions, and the tracking process is based on low-level pixel features. Similarly, an algorithm based on modeling the foreground and background regions with a mixture of Gaussians is proposed [30] where the target location is estimated by updating a level set function. However, only pixel features drawn from the mixture of segments are used rather than the mid-level segments. In [31], an offline segmentation and tracking algorithm is proposed to separate the moving object from the background by solving an optimization problem. Image data is represented by a multi-label Markov random field model and the optimization is carried out when the whole sequence is available. Recently, the Hough-based tracking (HT) algorithm [32] is developed for handling non-rigid objects based on online random forests and Hough voting of detected regions with verification from the support of foreground regions. Different from our work that constructs an appearance model by directly using superpixels as features, segmentation is used to coarsely separate the target from the background and update the classifiers [32].

In this work, we propose a tracking method from the perspective of mid-level vision with structural information captured in superpixels. By incorporating both appearance and spatial information, we construct a novel superpixel-based appearance model to separate the target from the background, thereby facilitating the proposed algorithm to handle heavy occlusion and recover from drifts for robust object tracking.

III. Proposed Algorithm

We present details of the proposed image representation scheme and tracking algorithm in this section. Our algorithm is formulated within the Bayesian framework in which the maximum a posterior estimate of the state given the observations up to time t is computed by

$$p(X_t|Y_{1:t}) = \alpha p(Y_t|X_t) \int p(X_t|X_{t-1})p(X_{t-1}|Y_{1:t-1})dX_{t-1}, \tag{1}$$

where X_t is the state at time t, $Y_{1:t}$ denotes all the observations up to time t, and α is a normalization term. In this work, the target state is defined as $X_t = (X_t^c, X_t^{sx}, X_t^{sy})$, where X_t^c represents the center location of the target, X_t^{sx} and X_t^{sy} denote its scale in x-axis and y-axis, respectively. As demonstrated by numerous works in the object tracking literature, it is critical to construct an effective observation model $p(Y_t|X_t)$ and an efficient motion model $p(X_t|X_{t-1})$.

In our formulation, a robust discriminative appearance model is constructed which, given an observation, computes the likelihood of it belonging to the target or the background. Thus the observation estimate of a certain target candidate X_t is proportional to its confidence:

$$p(Y_t|X_t) \propto \hat{C}(X_t), \tag{2}$$

where $\hat{C}(X_t)$ represents the confidence of an observation at state X_t being the target. The state estimate of the target $\hat{X}_t$ at time t can be obtained by the MAP estimate over the N samples at each time t. Let $X_t^{(l)}$ denote the l-th sample of the state X_t,

$$\hat{X}_t = \underset{X_t^{(l)}}{\text{argmax}}\, p(X_t^{(l)}|Y_{1:t}) \;\forall\, l = 1, \ldots, N. \tag{3}$$

In the following, the superpixel-based discriminative appearance model for tracking is introduced in Section III-A, followed by construction of the confidence map based on this model in Section III-B. The observation and motion models are presented in Section III-C, and then the update scheme.

A. Superpixel-Based Discriminative Appearance Model

To construct an appearance model for both the target and the background, prior knowledge regarding the label of each pixel can be learned from a set of m training frames. That is, for a certain pixel at location (i, j) in the t-th frame $pixel(t, i, j)$, we have:

$$y_t(i, j) = \begin{cases} 1 & \text{if } \; pixel(t,i,j) \in \text{target} \\ -1 & \text{if } \; pixel(t,i,j) \in \text{background}, \end{cases} \tag{4}$$

where $y_t(i, j)$ denotes the label of $pixel(t, i, j)$. Assume that the target object can be represented by a set of superpixels without significantly destroying the boundaries between target and background (i.e., only few superpixels contain almost equal amount of target pixels and background pixels), prior knowledge regarding the target and the background appearance can be modeled by

$$y_t(r) = \begin{cases} 1 & \text{if } \; sp(t, r) \in \text{target} \\ -1 & \text{if } \; sp(t, r) \in \text{background}, \end{cases} \tag{5}$$

where $sp(t, r)$ denotes the r-th superpixel in the t-th frame, and $y_t(r)$ denotes its corresponding label. However, such prior knowledge is not at our disposal in most tracking scenarios, and one feasible way to achieve this is to infer prior knowledge from a set of samples, $\{X_t\}_{t=1}^m$, prior to the tracking process starts. We present a method to extract similar information as Eq. 5 from a small set of samples.

First, we segment the surrounding region of the target in the t-th training frame into N_t superpixels. The surrounding region is a square area centered at the location of target X_t^c, and its side length is equal to $\lambda_s[S(X_t)]^{\frac{1}{2}}$, where $S(X_t)$ represents the area size of target area X_t. We use squared region for simplicity and it works well in practice although rectangular or more sophisticated regions can be used at the expense of using a larger state space. The parameter λ_s is a constant parameter, which controls the size of this surrounding region, and is set to 3 in all experiments. Therefore, the surrounding region is large enough to cover the entire object in the last frame and include sufficient background region around the object for better discrimination. Each superpixel $sp(t, r)$ $(t = 1, \ldots, m, \; r = 1, \ldots, N_t)$ is represented by a feature vector f_t^r (See Fig. 2(a)–(c)). The mean shift clustering algorithm, with a single parameter controlling the bandwidth of the kernel function, has been shown to better capture the relationship among superpixels rather than other methods (e.g., k-means). Thus, in this work we apply the mean shift clustering algorithm on the feature pool $F = \{f_t^r|t = 1, \ldots, m; \; r = 1, \ldots, N_t\}$ and obtain n different clusters.

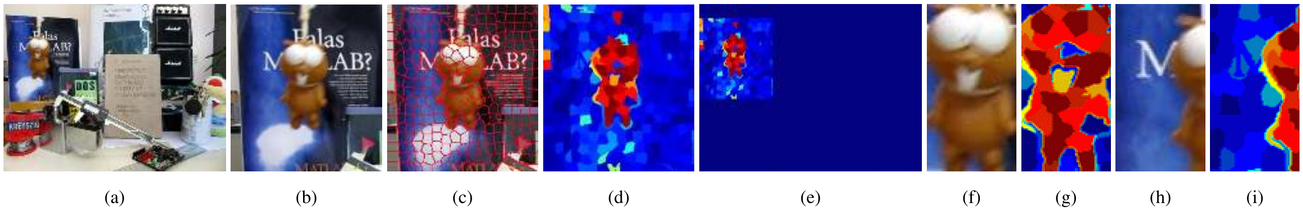


Fig. 2. Illustration of confidence map for state prediction. (a) A new frame at time t. (b) Surrounding region of the target in the last frame, i.e., at state $X_t^{(1)}$. (c) Superpixel segmentation from (b). (d) The computed confidence map of superpixels using Eq. 8 and Eq. 7. The superpixels colored with red indicate strong likelihood of belonging to the target, and those colored with dark blue indicate strong likelihood of belonging to background. (e) the confidence map of the entire frame. (f), (g) and (h), (i) show two target candidates with high and low confidence, respectively.

In the feature space, each cluster $clst(i)$ $(i = 1, \ldots, n)$ is represented by its cluster center $f_c(i)$, its cluster radius $r_c(i)$ and its own cluster members $\{f_t^r | f_t^r \in clst(i)\}$.

As every cluster $clst(i)$ corresponds to its own image region $S(i)$ in the training frames (image regions that superpixel members of cluster $clst(i)$ cover), we count two scores for each cluster $clst(i)$, $S^+(i)$ and $S^-(i)$. The former denotes size of cluster area $S(i)$ overlapping the target area at state X_t in the corresponding training frames, and the latter denotes the size of $S(i)$ outside the target area. Intuitively, the ratio $S^+(i)/S^-(i)$ indicates the likelihood that superpixel members of $clst(i)$ appear in the target area. Consequently, we assign each cluster a target-background confidence value between 1 and -1 to indicate whether its superpixel member belonging to the target or the background.

$$C_i^c = \frac{S^+(i) - S^-(i)}{S^+(i) + S^-(i)}, \ \forall\, i = 1, \ldots, n. \tag{6}$$

where larger positive values indicate high confidence to assign the cluster to target and vice versa.

Our superpixel-based discriminative appearance model is constructed based on four factors, cluster confidence C_i^c, cluster center $f_c(i)$, cluster radius $r_c(i)$ and cluster members $\{f_t^r | f_t^r \in clst(i)\}$, which are used for determining the cluster for a certain superpixel as described in the following sections. By applying the confidence values of each cluster to superpixels in the training frames, similar prior knowledge as Eq. 5 can be learned from a set of training images.

The merits of the proposed superpixel-based discriminative appearance model are illustrated in Fig. 4 and Section IV. Namely, few background superpixels that appear in the target area (as a result of drifts or occlusions) are likely to be clustered into the same group with other background superpixels. Thus the background pixels within the target region (enclosed by a rectangle) have negligible effect to our appearance model during training and update.

B. Confidence Map

When a new frame arrives, we first extract a surrounding region[2] of the target and segment it into N_t superpixels (See Fig. 2(b) and (c)). To compute a confidence map for the current frame, we evaluate every superpixel and compute its confidence value. The confidence value of a superpixel depends on two factors: the cluster it belongs to, and the distance between this superpixel and the corresponding cluster center in the feature space. The rationale for the first criterion is that if a certain superpixel belongs to cluster $clst(i)$ in the feature space, then the target-background confidence of cluster $clst(i)$ indicates how likely it belongs to the target or background. The second term is a weighting term that takes the distance metric into consideration. The farther the feature of a superpixel f_t^r lies from the corresponding cluster center $f_c(i)$ in feature space, the less likely this superpixel belongs to cluster $clst(i)$. The confidence value of each superpixel is computed as follows:

$$C_r^s = w(r, i) \times C_i^c, \ \forall\, r = 1, \ldots, N_t, \tag{7}$$

and

$$\begin{gathered} w(r, i) = \exp(-\lambda_d \times \tfrac{\|f_t^r - f_c(i)\|_2}{r_c(i)}), \\ \forall\, r = 1, \ldots, N_t, \quad i = 1, \ldots, n, \end{gathered} \tag{8}$$

where $w(r, i)$ denotes the weighting term based on the distance between f_t^r (the feature of $sp(t, r)$, the r-th superpixel in the t-th frame) and $f_c(i)$ (the feature center of the cluster that $sp(t, r)$ belongs to). The parameter $r_c(i)$ denotes the cluster radius of cluster $clst(i)$ in the feature space, and λ_d is a normalization term (set to 2 in all experiments). By taking these two terms into account, C_r^s is the confidence value for superpixel r at the t-th frame, $sp(t, r)$.

We obtain a confidence map for each pixel on the entire current frame as follows. Every pixel in the superpixel $sp(t, r)$ is assigned with confidence C_r^s, and every pixel outside this surrounding region with confidence value -1. Fig. 2(a)–(e) show the steps how the confidence map is computed with a new frame arriving at time t. This confidence map is computed based on our appearance model described in Section III-A. In turn, the following steps for identifying the likely locations of the target in object tracking are based on this confidence map.

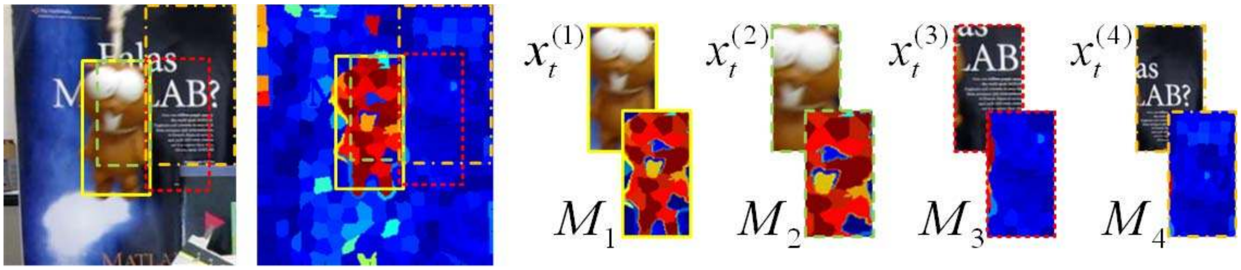
C. Observation and Motion Models

The motion (or dynamical) model is assumed to be Gaussian distributed,

$$p(X_t | X_{t-1}) = \mathcal{N}(X_t; X_{t-1}, \Psi), \tag{9}$$

where Ψ is a diagonal covariance matrix whose elements are the standard deviations for location and scale, i.e., σ_c and σ_s. The values of σ_c and σ_s dictate how the proposed algorithm accounts for motion and scale change.

[2] A square area centered at X_{t-1}^c with side length $\lambda_s[S(X_{t-1})]^{\frac{1}{2}}$.



warped image confidence map

Fig. 3. Confidence map. Four target candidate regions corresponding to states $X_t^{(i)}$, $i = 1, \ldots, 4$ are shown both in warped image and the confidence map. These candidates' confidence regions M_i, $i = 1, \ldots, 4$ have the same canonical size (upper right) after normalization. Based on Eq. 10, candidate $X_t^{(1)}$, $X_t^{(2)}$ have similar positive confidence C_1, C_2, and $X_t^{(3)}$, $X_t^{(4)}$ have similar negative confidence C_3, C_4. However, candidate $X_t^{(2)}$ covers less target area than $X_t^{(1)}$, and $X_t^{(4)}$ covers more background area than $X_t^{(3)}$. Intuitively, target-background confidence of $X_t^{(1)}$ should be higher than $X_t^{(2)}$, while confidence of $X_t^{(4)}$ should be lower than $X_t^{(3)}$. These two factors are considered in computing confidence map as described in Section III-C.

We normalize all these candidate image regions into canonical sized maps $\{M_l\}_{l=1}^N$ (the size of the target corresponding to X_{t-1} is used as the canonical size). We denote $v_l(i, j)$ as the value at location (i, j) of the normalized confidence map M_l of $X_t^{(l)}$, and then we accumulate $v_l(i, j)$ to obtain the confidence C_l for the state $X_t^{(l)}$,

$$C_l = \sum_{(i,j)\in M_l} v_l(i, j). \tag{10}$$

However, this target-background confidence value C_l does not take scale change into account. In order to make the tracker robust to the scale change of the target, we weigh C_l with respect to the size of each candidate as follows:

$$\hat{C}_l = C_l \times [S(X_t^{(l)})/S(X_{t-1})],\ \forall\, l = 1, \ldots, N, \tag{11}$$

where $S(X_{t-1})$ represents the area size of target state X_{t-1} and $S(X_t^{(l)})$ represents the area size of a candidate state $X_t^{(l)}$. For the target candidates with positive confidence values (i.e., indicating they are likely to be targets), the ones with larger area size should be weighted more. For the target candidates with negative confidence values, the ones with larger area size should be weighted less. This weighting scheme ensures our observation model $p(Y_t|X_t^s)$ is adaptive to scale change. Fig. 3 illustrates this weighting scheme.

We normalize the final confidence of all targets $\{\hat{C}_l\}_{l=1}^N$ within the range of $[0, 1]$ for computing likelihood of $X_t^{(l)}$ for our observation model:

$$p(Y_t|X_t^{(l)}) = \underline{\hat{C}_l},\ \forall\, l = 1, \ldots, N, \tag{12}$$

where $\underline{\hat{C}_l}$ denotes the normalized confidence value for each sample. With the observation model $p(Y_t|X_t^{(l)})$ and the motion model $p(X_t^{(l)}|X_{t-1})$, the MAP state $\hat{X}_t$ can be computed with Eq. 3. Fig. 2 (f)–(i) show two samples and their corresponding confidence maps. As shown in these examples, the confidence maps facilitate the process of determining the most likely target location.

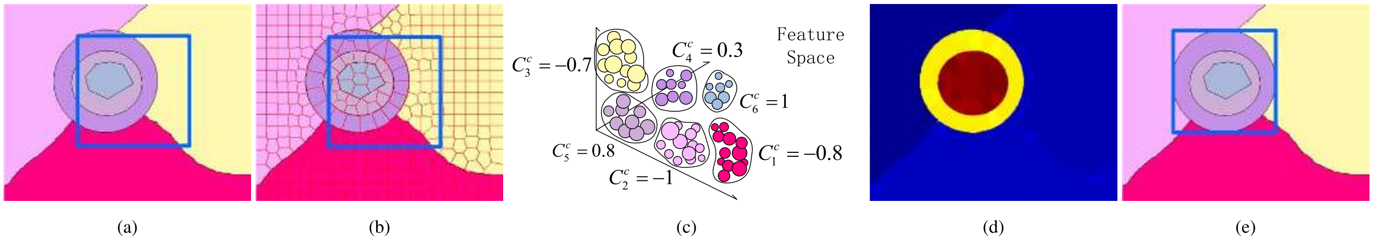
D. Online Update With Occlusion and Drifts

We apply superpixel segmentation to the surrounding region of the target (rather than the entire image) for efficient and effective object tracking. An update scheme with sliding window is adopted, in which a sequence of H frames is stored during the tracking process. For every U frames, we add a new frame into this sequence, and delete the oldest one. That is, this process retains a record of the past $H \times U$ frames. For each frame in this sequence, the estimated state $\hat{X}_t$ and the result of superpixel segmentation are saved. We update the appearance model with the retained sequence every W frames in the same way as that of the training phase described in Section III-A.

With the proposed discriminative appearance model using mid-level cues, we present a simple but efficient method to handle occlusion in object tracking. The motivation is that the confidence values of target estimates in the retained sequence capture the most recent appearance information of the target object in a short period which can be used to measure the quality of current MAP estimate. For a state $X_t^{(l)}$ at time t, its confidence C_l (from Eq. 10) is bounded within a range, $[-S(X_t^{(l)}), S(X_t^{(l)})]$. The upper bound indicates that all pixels in the image region corresponding to $X_t^{(l)}$ are assigned with the highest confidence of belonging to the target, and conversely the lower bound indicates that all pixels belong to the background. We compute an occlusion indicator, O_t, and determine whether it is above a threshold θ_o to detect heavy or full occlusions:

$$O_t = \mu_C - \frac{\max(\{C_l\}_{l=1}^N)}{S(X_{t-1})}, \tag{13}$$

where μ_C is the average of normalized confidence (from Eq. 10) of the target estimates in the retained sequence of H frames. The denominator is a normalization term to ensure the range of the second term is $[-1,1]$. The formula reflects the difference between the normalized confidence C_l of the MAP estimate of current frame and the average normalized confidence of targets in the retained sequence. Without any prior information, the average of confidence of previous target estimates in a short period is the most reliable value. Therefore, large difference indicates a small confidence value of the current MAP estimate which is likely caused by occlusion. If the confidence C_l of the MAP estimate at the current frame



(a) (b) (c) (d) (e)

Fig. 4. Recovering from drifts. (a) A target object with tracking drifts. (b) The surrounding region of the target is segmented into superpixels. (c) Clustering results of (b) in the feature space and the target-background confidence of each cluster. (d) The confidence map in a new frame computed with clustering results. (e) The MAP estimate of the target area where the tracker recovers from drifts. This illustration shows even if our tracker experiences drifts during tracking (See (a)), our appearance model obtains sufficient information from surrounding background area by update, and provides the tracker with more discriminative strength against drifts than holistic appearance models.

is much less than the average of normalized confidence of the retained sequence, it means that the candidate region is very likely to belong to background area, and a heavy occlusion is deemed to occur.

As the target object is considered being occluded, the target estimate X_{t-1} of the last frame is considered as the target estimate $\hat{X}_t$ for the current frame. Furthermore, instead of deleting the oldest frame when we add one new frame to the end of the retained sequence, we delete the k-th (e.g., $k = 8$, $k < H$) frame of the sequence. In this manner, our tracker does not remove all appearance information of target object when long-duration occlusion occurs, and meanwhile does not continue to learn from occluded examples. Without this update mechanism, our tracker may update with wrong examples when the target object is occluded or un-occluded. As demonstrated by our experiments presented in Section IV, robust tracking results can be obtained with this scheme.

The confidence map with update is also used to recover from drifts in our algorithm. Fig. 4 illustrates how the proposed method recovers from drifts using the information from superpixels and the confidence map. As the superpixel segmentation is carried out at frame t (within the image regions where the target appears at frame $t - 1$), the computed confidence map provides strong evidence where the target will appear, thereby correcting potential drifts from inaccurate state estimate at frame $t - 1$. The main steps of the proposed algorithm are summarized in Fig. 5.

```
Initialization:
for $t = 1$ to $m$ (e.g., $m$ is set to $4$ in all experiments)
    1) Initialize parameters of our algorithm in frame $t$.
    2) Segment the surrounding region of $X_t$ into $N_t$ superpixels, and extract their features $\{f_t^r\}_{r=1}^{N_t}$ for training.
end
Obtain a feature pool $F = \{f_t^r | t = 1, ..., m;\ r = 1, ..., N_t\}$. Apply mean shift clustering and obtain the superpixel-based discriminative appearance model by Eq. 6.

Tracking:
for $t = m + 1$ to the end of the sequence
    1) Segment a surrounding region of $X_t$ into $N_t$ superpixels and extract their features. Compute the target-background confidence map using Eq. 8 and Eq. 7.
    2) Sample $N$ candidate states $\{X_t^{(l)}\}_{l=1}^{N}$ with the confidence map.
    3) Compute motion parameters $p(X_t^{(l)} | X_{t-1})$ by Eq. 9 and their likelihoods $p(Y_t | X_t^{(l)})$ by Eqs. 10-12.
    4) Estimate MAP state $\hat{X}_t$ using Eq. 3.
    5) Detect heavy or full occlusion with Eq. 13.
    6) Add one frame into the update sequence every $U$ frames.
    7) Update the appearance model every $W$ frames.
end
```

Fig. 5. Main steps of the proposed superpixel tracking algorithm.

IV. Experiments

We present the experimental setups and extensive empirical results using challenging image sequences as well as observations in this section. Our algorithm is implemented in MATLAB and runs at 5 frames per second on a 3.4 GHz CPU with 12 GB memory. The most time-consuming part is due to the use of the mean shift clustering algorithm. The MATLAB source code and datasets are available at http://www.umiacs.umd.edu/∼fyang/spt.html.

A. Experimental Setups

We use a normalized histogram in the HSI color space as the feature for each superpixel. The HSI color space reduces the effect of lighting change on pixels and empirically shows more discriminative ability in distinguishing different superpixels than other color spaces. Therefore, the superpixels using HSI color space can better differentiate the foreground from the background. The SLIC algorithm [28] is applied to extract superpixels where the spatial proximity weight and number of superpixels are set to 10 and 300, respectively. The bandwidth of the mean shift clustering [33] is set to 0.18. We note that the bandwidth needs to be wide enough to separate superpixels from the target and background into different clusters. The parameters of segmentation (i.e., number of superpixels and spatial proximity weight) and clustering (i.e., bandwidth of kernel function) are empirically determined from the results on a randomly selected sequence and fixed for all the test videos.

To collect a training dataset in the initialization step, the target regions in the first 4 frames are either located by an object detector or manually cropped. The parameters (H, U, and W) are empirically determined and fixed for all test sequences (i.e, they are set to 15, 1, and 10 in all the experiments). The parameters, σ_c and σ_s, of Eq. 9 are set to 7.6 and 7 in anticipation of the fastest motion speed or scale change of the target objects. In occlusion detection, μ_C is set to 0.5 for simplicity. The occlusion detection threshold θ_o is empirically defined as 0.515 and fixed for all sequences. The update parameters and the occlusion detection threshold are important to the proposed tracker since they control how much new information can be used for updating the tracker.

We evaluate the proposed algorithm on 12 challenging sequences where 6 of them have been tested extensively in prior work [5], [8], [12], [18] and the others are collected on our own. These sequences include most challenging factors in visual tracking: complex background, moving camera, fast movement, large variation in pose and scale, half or full occlusion, shape deformation and distortion (see Figs. 1, 8, 9).

The proposed superpixel-based tracking (SPT) algorithm is evaluated against several state-of-the-art tracking methods, including the IVT [3], Frag [5], MIL [7], ℓ_1 [11], PROST [18], VTD [12], TLD [13], Struck [14] and HT [32] tracking methods. Since we use color features in our method, we compare the SPT method with two popular color-based trackers, the mean shift tracker with adaptive scale (MS) [34] and the adaptive color-based particle filter (PF) [35] method. In addition, our work can be easily extended to segment salient foreground target from background, and the relevant results are presented in Section IV-C.

B. Empirical Results

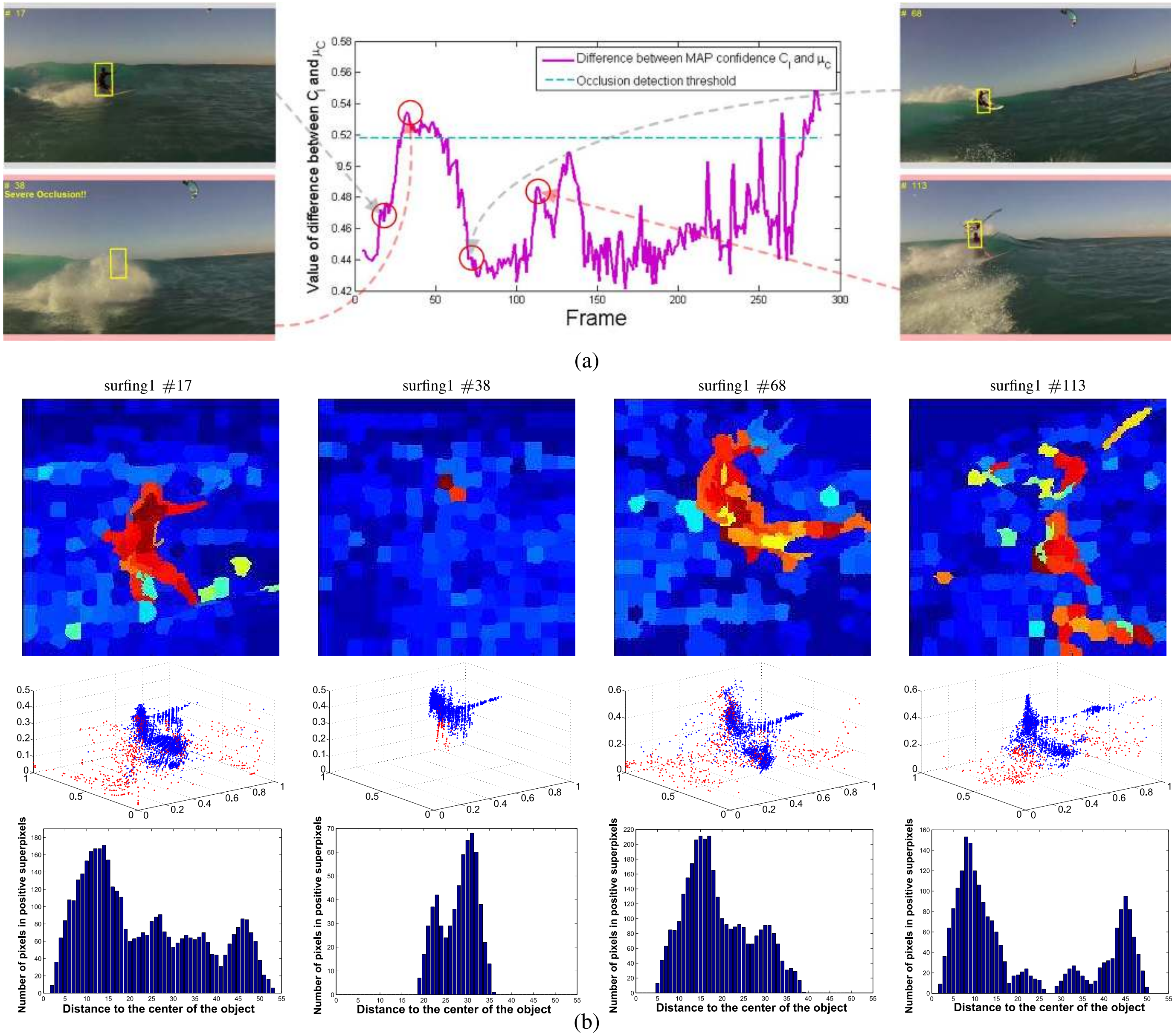
We first evaluate our algorithm with the sequences used in prior works: *singer1* and *basketball* from the VTD method [12], *transformer* from the PDAT tracker [8], *lemming* and *liquor* from the PROST algorithm [18], and *woman* from the Frag approach [5]. We then test on 6 sequences from our own dataset: *bolt*, *bird1*, *bird2*, *girl*, *surfing1* and *racecar*. Experimental results show that the proposed method with fixed parameters performs well in all sequences with various challenging factors for object tracking. For fair comparisons, we carefully adjust the parameters of every tracker with the code provided by the authors and present the best results from 5 runs, or taken directly from the presented results in prior works. All the tracking results can be found at http://www.umiacs.umd.edu/∼fyang/spt.html.

Effectiveness of occlusion handling: We first demonstrate the effectiveness of the proposed appearance model in handling occlusions using an example shown in Fig. 6. In this sequence, we aim to track the kitesurfer in the sequence *surfing1* whose appearance varies significantly due to occlusion and pose change. In Fig. 6(a), we show a plot of occlusion indicator O_t (from Eq. 13) with four representative results. The plot of occlusion indicator O_t describes different situations during tracking. When the target object is visible (frame #17 and #68), the value of O_t is low to ensure normal updates as no occlusion occurs. In contrast, the value of O_t is high when the target object is occluded. In frame #113, the target object is partially occluded by the wave and the indicator value is larger than those in frame #17 and #68. When the object is heavily occluded (frame #38), the indicator value is larger than the predefined threshold and the appearance model is not updated with the current observation. With this update mechanism, our tracker overcomes the problem of accumulating errors in model update which leads to tracking failure.

In Fig. 6(b), the corresponding confidence maps of local regions are shown in the first row. The target object can be easily identified when there is no occlusion. The HSI color distributions of the pixels within the corresponding local regions are presented in the second row where pixels belonging to the foreground and background are represented by red and blue points, respectively. It is clear that when heavy occlusion occurs, there are few foreground pixels and many background ones. Due to the discriminative formulation of the proposed superpixel-based appearance model which adaptively learns both the foreground and the background, our method can better find the target object which is salient in the confidence map. Our appearance model also implicitly utilizes the spatial information to ensure tracking accuracy. In this sequence, the color distributions of frame #68 and #113 are similar in terms of foreground and background pixels. However, the scenes and object appearances are very different since some of the identified foreground pixels in frame #113 are from another surfer next to the target surfer. In this case, color distributions alone do not carry sufficient information for object tracking. Fig. 6(b) shows the spatial distributions of distance to the region center for foreground pixels (i.e., the spread of the foreground pixels). For tracking, our algorithm favors the sample with a large amount of foreground pixels close to the center of the local region (i.e., compact potential target region). The histogram modes show that a significant number of foreground pixels appear at a distance from the center of the local region in frame #113 compared to those in frame #68 although they exhibits similar patterns in the color distributions. Since the foreground pixels close to the object center are usually more important than others which usually can be enclosed by convex shapes, the pixels away from the region center are weighted less in our algorithm (See Eq. 7). By learning the color features of both foreground and background and utilizing spatial information, the proposed appearance model is able to deal with challenging factors in object tracking, as shown in the following sections.

The proposed occlusion handling process requires a proper threshold θ_o to be defined empirically. While the proposed occlusion handling mechanism performs well in most scenarios (as shown in the experiments on different scenarios using a fixed threshold), it is likely to fail when objects are fully occluded for a long period of time, which is a common problem for almost all online tracking algorithms.

Comparison with color-based trackers: We compare the proposed SPT algorithm with two color-based trackers, the mean shift tracker with adaptive scale (MS) [34] and the adaptive color-based particle filter (PF) tracker [35]. As shown in Fig. 7, the PF tracker does not perform well when objects



(a)

(b)

Fig. 6. Illustration of effective occlusion handling and proposed appearance model. (a) Plot of occlusion indicator (Eq. 13) with four representative tracking examples, which contain partial or severe occlusions. The plot indicates whether the target object is occluded or not. (b) First row: the corresponding confidence maps of the four examples, where red color indicates high probability of a pixel belonging to the foreground while blue color indicates high probability of a pixel belonging to the background. Second row: the corresponding HSI color distributions. The axes represent three color channels and pixels belonging to foreground and background are shown in red and blue points. Third row: histograms of spatial distributions of foreground pixels. The x-axis and y-axis indicate the distance of a pixel to the center of the local region and the number of pixels lying within a small distance range. Note that the histograms are significantly different for frame #68 and #113 although they have similar color distributions (best viewed on high-resolution display).

appear in cluttered backgrounds, with drastic movements or under heavy occlusion. While it performs well in the *girl* and *racecar* sequences, it fails in all other videos. The MS tracker does not perform well when the target object undergoes a large appearance change due to non-rigid motion, lighting change and heavy occlusion (Fig. 7). While this MS tracker is designed to account for scale change, it is less effective in dealing with lighting variation and occlusion.

On the other hand, the discriminative appearance model based on mid-level representation of our SPT method alleviates negative influences from noise and background clutters. Consequently, our tracker is able to track objects undergoing heavy occlusion, non-rigid deformation and lighting change in cluttered backgrounds (Fig. 7).

Comparison with other state-of-the-art trackers: The quantitative comparisons are presented in Tables I and II. Table II shows quantitative comparisons based on evaluation metric of the PASCAL VOC object detection [36], which is also used in other tracking algorithm [18]. For fair comparisons, we use elliptical target area for the mean shift tracker and the adaptive color-based particle filter tracker, and rectangular bounding box for the HT tracker to calculate the metric used in PASCAL

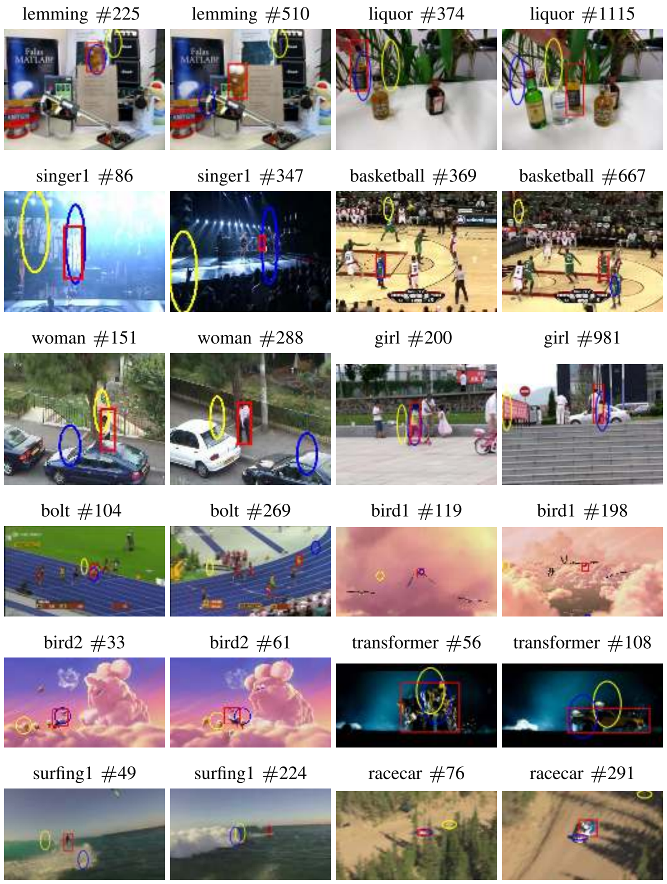


Fig. 7. Tracking results with comparisons to color-based trackers. The results by the MS tracker, the PF method and our algorithm are represented by yellow ellipse, blue ellipse and red rectangles. The proposed tracker is able to handle cluttered background (*girl* and *basketball*), pose change (*liquor*, *bolt*, *bird2* and *surfing1*), scale change (*singer1* and *racecar*), heavy occlusion (*lemming*, *liquor*, *woman*, *bird1*, *surfing1* and *racecar*), shape deformation (*transformer*) and lighting condition change (*singer1*).

TABLE I

CENTER LOCATION ERROR

Sequence	MS	PF	IVT	Frag	MIL	PROST	VTD	ℓ_1	TLD	Struck	HT	SPT
lemming	236	184	**14**	84	**14**	23	98	182	104	134	118	**7**
liquor	137	28	238	31	165	**22**	155	80	28	124	202	**9**
singer1	116	25	**5**	20	20	–	**3**	**3**	**5**	16	52	**5**
basketball	203	21	120	14	104	–	**11**	100	170	153	19	**6**
woman	32	79	133	112	120	–	109	113	95	**5**	122	**11**
transformer	46	49	130	47	33	–	43	108	**23**	54	31	**14**
bolt	204	34	382	100	380	–	**14**	369	90	387	373	**6**
bird1	330	**137**	230	223	270	–	250	226	77	148	203	**47**
bird2	73	75	119	28	18	–	50	19	86	88	**10**	**17**
girl	304	**16**	184	106	55	–	57	177	151	119	232	**10**
surfing1	81	156	141	199	319	–	84	228	**27**	265	287	**48**
racecar	199	**11**	340	94	104	–	196	224	134	202	228	**4**

VOC tests. In addition, the tracking error plots in terms of center position are shown in the supplementary document. We note that the results on two sequences are taken directly from [18] as the source code is not available for evaluation.

Figs. 8 and 9 show some screenshots of the tracking results. For clarity of presentation, only the top four trackers with the lowest center location errors are shown and more results are presented in the supplementary document. In the following, we evaluate these algorithms in terms of challenging factors in object tracking.

TABLE II

NUMBER OF SUCCESSFULLY TRACKED FRAMES

Sequence	MS	PF	IVT	Frag	MIL	PROST	VTD	ℓ_1	TLD	Struck	HT	SPT
lemming	171	426	1053	680	**1105**	1100	470	226	361	652	281	**1277**
liquor	413	1202	400	1377	353	**1444**	471	988	1398	405	1	**1689**
singer1	64	96	328	87	87	–	**351**	**351**	**351**	87	79	**347**
basketball	78	455	80	512	204	–	**619**	30	46	85	433	**695**
woman	35	31	49	44	39	–	27	53	36	**333**	48	**298**
transformer	28	32	29	39	30	–	47	33	43	34	**73**	**124**
bolt	15	172	5	33	12	–	**199**	14	49	9	4	**231**
bird1	1	6	4	44	**118**	–	7	7	25	17	**132**	84
bird2	36	19	9	44	**86**	–	9	76	12	14	83	**90**
girl	79	**1106**	107	632	575	–	832	391	169	246	1	**1439**
surfing1	36	16	24	28	10	–	24	22	**116**	24	25	**98**
racecar	43	**207**	17	111	33	–	42	59	24	51	56	**345**

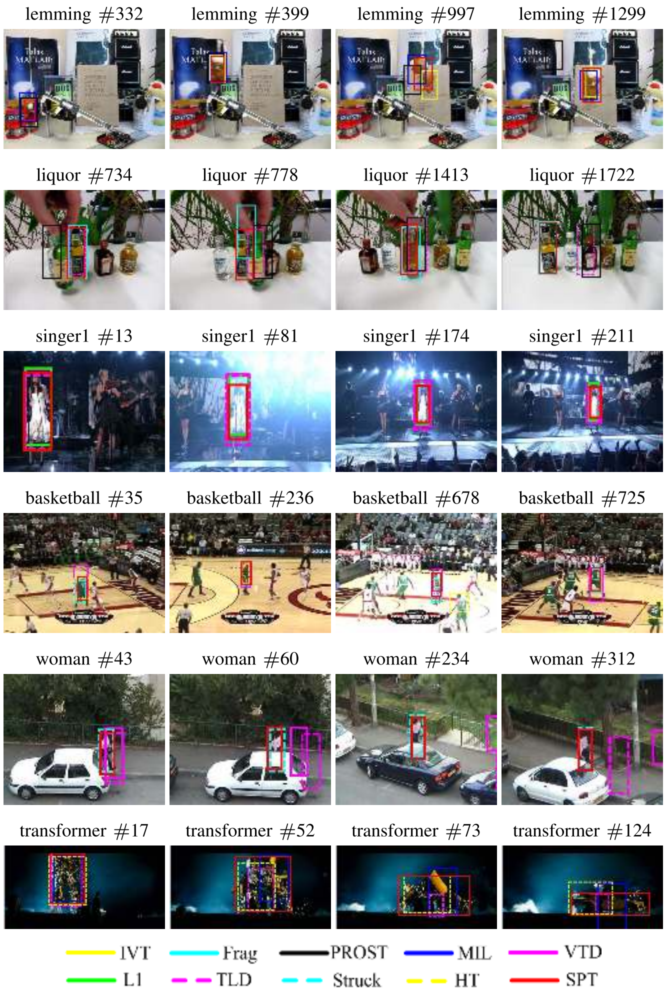


Fig. 8. Tracking results on the public datasets by the IVT, Frag, MIL, PROST, VTD, TLD, Struck, HT and SPT methods. The best four trackers in terms of errors of center location are shown.

Tracking drifts: While trackers based on holistic appearance models are able to track objects in many scenarios, they are less effective in handling drifts. The main reason is that these trackers typically focus on learning target appearance rather than the background (i.e., with a generative approach). Not equipped to discriminate the foreground from the background, these trackers usually do not recover from drifts as a result of accumulated tracking errors.

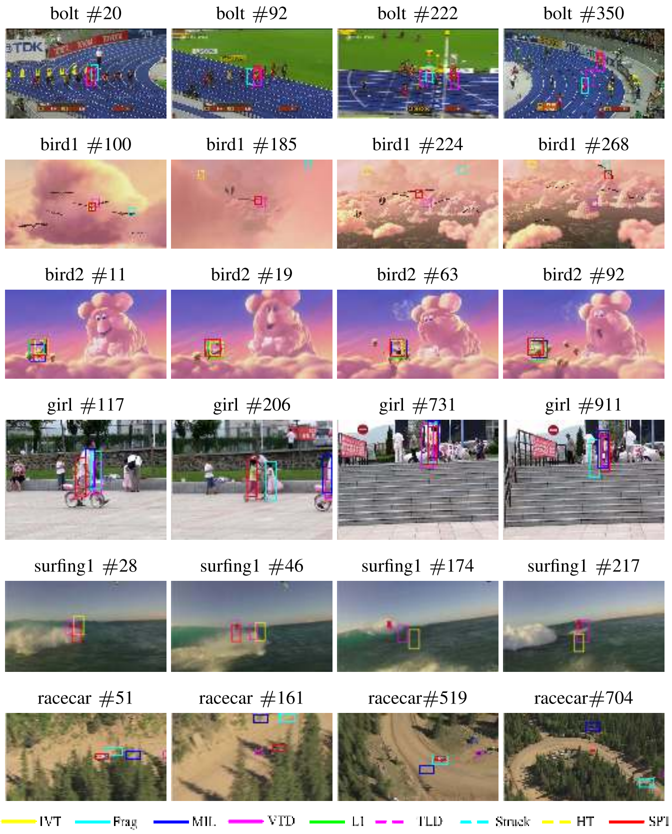


Fig. 9. Tracking results on our own datasets by the IVT, Frag, MIL, VTD, ℓ_1, TLD, Struck, HT and SPT methods. The best four trackers in terms of errors of center location are shown.

In the *basketball* sequence, the IVT and ℓ_1 trackers drift away from the target object when it undergoes non-rigid shape deformation and large pose change. Although with the mechanism of learning and classifying the features of both foreground and background, the MIL, Frag, TLD, HT and Struck trackers do not accurately locate the target object all the time. These tracking algorithms drift to background area for that they are not designed to account for non-rigid shape deformation and large pose change. Although the VTD tracker achieves the second best results in this sequence, its tracking results are not as accurate as ours. The reason is that it does not distinguish the target from the background, and considers some background pixels as parts of the target, thereby rendering inaccurate tracking results. In contrast, the discriminative appearance model of our tracker utilizes background information effectively and alleviates the tracking drift problem throughout this sequence. Similarly, the results from the *bolt* and *girl* sequences show that the proposed SPT tracker is able to recover from tracking drifts.

Fig. 9 shows the results using the *bird2* sequence. All other trackers except for the MIL, ℓ_1, HT and the proposed SPT methods have the tracking drift problem throughout the entire sequence. In addition, the SPT algorithm performs better than the MIL and ℓ_1 trackers in terms of accuracy. Although the HT method produces the lowest center location error, the proposed SPT tracker performs favorably in terms of the count of successfully tracked frames.

Pose variation and scale change: We evaluate all the trackers on sequences where objects undergo large pose and scale changes. First, we compare the performance of trackers in terms of handling large pose change. The *lemming* sequence in Fig. 8 shows that the IVT, MIL and PROST algorithms perform well as the methods with holistic appearance models are effective for tracking rigid targets (one component of the PROST method uses a static template) when there is no large change in scale and pose (e.g., out-of-plane rotation). However, their holistic appearance models are not effective in accounting for appearance change due to large pose change. On the other hand, our tracker is more robust when the target object undergoes pose variation due to the use of mid-level appearance model, and outperforms other trackers as the proposed discriminative appearance model learns the difference between the target and background with updates. In contrast, the VTD, TLD, Struck and HT algorithms, which use low-level features (i.e., Haar-like features, edge information, intensity values or gradients of pixels) rather than mid-level features, do not perform well even though these methods update the templates or use multiple classifiers (i.e., VTD, Struck and HT methods).

In the *bolt* sequence of Fig. 9, the appearance of the athlete changes significantly due to fast motion and camera view change. Most trackers fail to locate the athlete throughout the entire sequence except the VTD and SPT algorithms. In addition, the SPT method performs better than the VTD method in terms of the accuracy and successfully tracked frames. The *surfing1* sequence of Fig. 9 shows the results on where the appearance of the kitesurfer changes significantly due to fast motion, change of scale and pose, as well as occlusion. Despite large appearance change, the SPT algorithm is able to track the target object while the other methods do not perform well right in the beginning or the middle of the sequence.

We evaluate the tracking algorithms in terms of handling scale change. In the *singer1* sequence of Fig. 8, the target object undergoes large scale change due to camera movements. We note that this video is rather challenging as there also exists large lighting variation. Without adaptive scale change, the results of the Frag, MIL and Struck methods are less accurate although the targets are located within the tracking windows. The VTD, ℓ_1 and SPT methods adapt the tracking windows well according to the change of the target size, and perform equally well. In the *surfing1* sequence where the target object undergoes significant scale change, the SPT tracker outperforms all other evaluated trackers except the TLD method.

The target vehicle in the *racecar* sequence undergoes large scale change due to zooming movements of the camera. In addition to scale change and occlusions, this sequence is rather challenging as the aspect ratio of the racecar appears differently due to object motion and camera movements. Our SPT tracker is able to keep track of the racecar with tracking windows adjusted to the appearance change, thereby generating accurate results (Table I). In contrast, other trackers do not handle the scale and aspect ratio change well although they use several trackers or multiple classifiers.

These experimental results show that the foreground target object can be better separated from the background with the proposed discriminative mid-level representation scheme, thereby enabling the proposed SPT algorithm to achieve high tracking accuracy.

Shape deformation: The *transformer* sequence in Fig. 8 shows one example when drastic shape deformation occurs, tracking algorithms using holistic appearance models or blobs are unlikely to perform well (e.g., IVT, MIL, Frag and VTD). Other trackers such as the ℓ_1, TLD and Struck methods also generate inaccurate results and their tracking windows cover a small portion of the object with a few number of successfully tracked frames. Designed to handle non-rigid objects, the HT tracker is able to track the target object in this sequence. Nevertheless, tracking results are less accurate (Tables I and II).

The kitesurfer in the *surfing1* sequence also undergoes large deformation as he performs acrobatic movements. As mentioned above, the SPT method performs well in locating the kitesurfer in this sequence, which shows the robustness of the SPT method in dealing with both pose change and shape deformation at the same time. Our appearance model utilizes information of both target and background on local mid-level cues, and distinguishes target parts from background blocks precisely. As we cluster superpixels with similar HSI color features rather than modeling the holistic appearance of objects, the proposed tracker is not sensitive to the shape changes. When large shape deformation occurs, our tracker still can find the object as long as there exist a sufficient amount of superpixels belonging to the foreground cluster for finding their distinct characteristics. Thus, the proposed SPT tracker is able to generate the most accurate results.

Occlusion: As shown in Fig. 8, the target in the *liquor* sequence undergoes heavy occlusions several times. Since our superpixel-based discriminative appearance model is able to alleviate influence from background pixels and exploits both the target and background appearance, our tracker is able to detect and handle heavy occlusions accordingly (see Fig. 6 for more illustration). Although the PROST method may recover from drifts after occlusion, it does not succeed all the time. Furthermore, the other trackers fail as they are not able to handle large appearance change due to heavy occlusion or recover from drifts.

In the *woman* sequence of Fig. 8, the woman is partially occluded by sedans of different colors. Only the Struck and SPT methods successfully keep track of the woman and generate more accurate results than the other trackers. In the *girl* sequence of Fig. 9, the girl is occluded by another person for a few frames, which results in short-term full occlusion. However, the SPT method is able to track this girl and generates the best results in terms of accuracy and success rate. Other trackers drift away from the girl or only track the girl with inaccurate windows.

In the *surfing1* and *racecar* sequences, the short-term full occlusions also occur. The kitesurfer and the racecar are occluded by waves and trees, respectively. The SPT method performs well in handling such full occlusions as it quickly finds the object again after the occlusion without drifting. Among all other evaluated algorithms, only the TLD method locates the surfer in the *surfing1* sequence and only the PF tracker keeps track of the racecar in the entire *racecar* sequence. The TLD method re-detects the object in the frame when the tracking fails and handles the occlusion problem to some extent. In addition, the success of the PF method in the *racecar* sequence can be attributed to the fact that the target object can be distinguished from the background with adaptive color-based appearance model and particle filters. However, this adaptive method does not work well in other sequences due to the limitation of the generative color-based representation (which does not exploit background information).

The *bird1* sequence of Fig. 9 contains long time full occlusion where the bird is occluded by the cloud for nearly 60 frames. It is challenging for a tracker to find the bird again after such a long-term full occlusion, in addition to large pose change as it swings its wings. Only the proposed SPT algorithm is able to keep track of the bird. Because of the adaptive update strategy, the appearance model is not updated when heavy occlusion occurs and thus alleviates the potential problem of introducing inaccurate background information during updates.

Camera motion: The *surfing1* sequence of Fig. 9 contains significant camera shake in several frames as it is acquired by another person on a boat. The SPT method tracks the kitesurfer well despite the challenging factors including large pose and scale change, camera shake, fast motion, as well as occlusion.

C. Segmentation

Since superpixels are commonly used in segmentation, we demonstrate that the proposed discriminative appearance model can be used to separate the target object from the background in a frame. In this section, we present experimental results to show that video segmentation is also achieved as a by-product of the proposed tracking method.

Fig. 10 shows the tracking results of foreground and background segmentation from the *liquor* sequence where the original images, the confidence maps of the corresponding local regions (obtained using the appearance model), the foreground/background segmentation results and the tracking results are presented. The segmentation results are generated by adopting a simple adaptive threshold on the confidence map.

The segmentation results at frame #278 show that the target appearance is well modeled by our appearance model as the bottle is separated from the background. A simple adaptive threshold on the confidence map (second row) on the local region of the target object generates segmentation result (third row). We note that the target is well segmented from the rectangular region with only few pixels from the background. In fact, the parts of the target object are salient in the confidence map, which demonstrate the effectiveness of the proposed discriminative appearance model.

Frames #768 and #1287 of Fig. 10 show examples where the target object is partially and fully occluded. The confidence maps (second row) show that the proposed appearance model

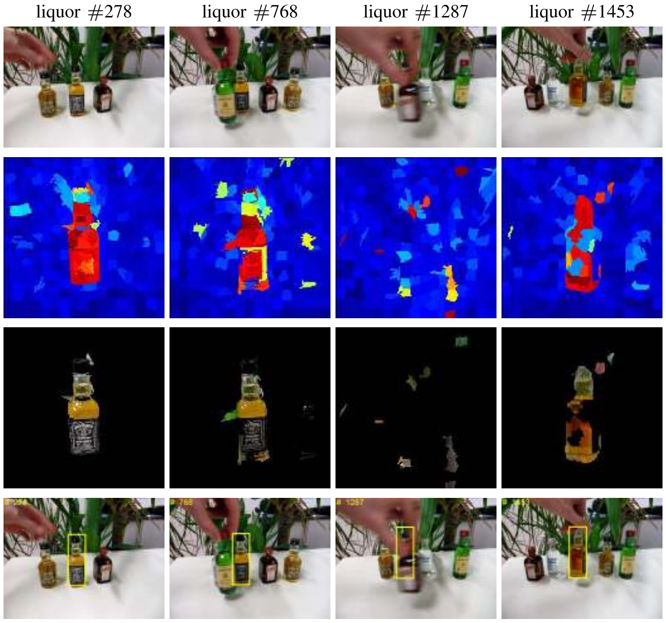


Fig. 10. Results of foreground/background segmentation and tracking across frames on sequence *liquor*. First row: original images. Second row: confidence maps of corresponding local regions, which is obtained by using the appearance model. Third row: the segmentation results. Fourth row: the final tracking results of each frame.

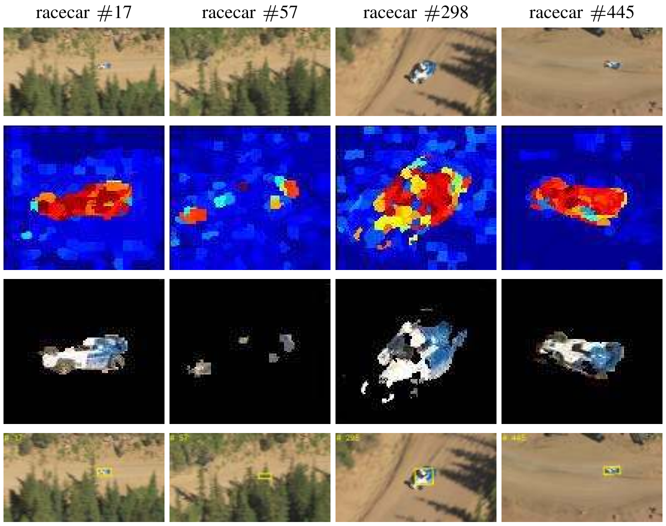


Fig. 11. Results of foreground/background segmentation and tracking across frames on sequence *racecar*. First row: original images. Second row: confidence maps of corresponding local regions, which is obtained by using the appearance model. Third row: the segmentation results. Fourth row: the final tracking results of each frame.

gives high confidence to the superpixels belonging to target object when the object is visible in the scene. On the other hand, the proposed confidence map gives low confidence values to superpixels of the same region when heavy occlusion occurs. In addition, the proposed appearance model is not updated with the occluding superpixels when occlusion is detected. The corresponding segmentation results also indicate that the foreground object is not visible in such cases. Since the target object is considered heavily occluded, the tracking window remains at the same position (fourth row) until it reappears in the scene.

The results at frame #1453 shows one example that our tracker is robust to pose change of the target object. With the proposed adaptive method, our tracker updates the the superpixel clusters and the discriminative appearance model, thereby dealing with pose change with accurate tracking and segmentation results when the bottle is rotated numerous times. Note that the appearance of the target is much different from that in previous frames (as a result of fast spinning), and our method is able to track and segment the object well (only a few superpixels belonging to the target are missing).

Fig. 11 shows another video segmentation results of the *racecar* sequence. Although the size of the racecar is small, especially there are only a number of pixels in frame #445, our method is able to find representative parts of the car and segment it from the background for tracking.

V. Conclusion

In this paper, we propose a robust tracker based on a discriminative appearance model and superpixels. We show that the use of superpixels provides flexible and effective mid-level cues, which are incorporated in an appearance model to distinguish the foreground target and the background. The appearance model is constructed by clustering a number of superpixels into different clusters. During tracking, we segment a local region around the target into superpixels and assign them confidence values to form a confidence map by computing the distance between a superpixel and the clusters. The proposed appearance model is used for object tracking to account for large appearance change due to shape deformation, occlusion and drifts. Numerous experimental results and evaluations demonstrate the SPT tracker performs favorably against existing state-of-the-art algorithms in the literature in handling various situations, such as large variation of pose and scale, shape deformation, occlusion and camera shake.

We demonstrated that the SPT method is able to segment the target objects out during tracking. These segmentation results can be further improved by refinement processes or labeling methods with spatio-temporal information. As the most time consuming part of the proposed algorithm is the mean shift clustering method, we will explore other efficient and effective alternatives. As simple HSI color features are used in the proposed tracking method, better features can be incorporated to further improve the tracking results.

References

[1] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. 415–422.

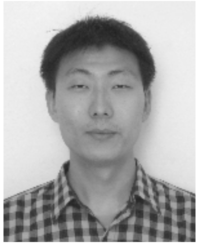
[2] R. Collins and Y. Liu, "On-line selection of discriminative tracking features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 346–352.

[3] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang, "Incremental learning for visual tracking," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2005, pp. 793–800.

[4] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 260–267.

[5] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 798–805.

[6] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised on-line boosting for robust tracking,” in *Proc. 10th Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 234–247.
[7] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 983–990.
[8] J. Kwon and K. M. Lee, “Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping Monte Carlo sampling,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1208–1215.
[9] B. Han, Y. Zhu, D. Comaniciu, and L. S. Davis, “Visual tracking by continuous density propagation in sequential Bayesian filtering framework,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 919–930, May 2009.
[10] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, “SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2937–2944.
[11] X. Mei and H. Ling, “Robust visual tracking using l1 minimization,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1436–1443.
[12] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1269–1276.
[13] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N learning: Bootstrapping binary classifiers by structural constraints,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 49–56.
[14] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” in *Proc. IEEE ICCV*, Nov. 2011, pp. 263–270.
[15] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.
[16] K. Cannons, “A review of visual tracking,” Dept. Comput. Sci., York Univ., Toronto, ON, Canada, Tech. Rep. CSE-2008-7, 2008.
[17] M.-H. Yang and J. Ho, “Toward robust online visual tracking,” in *Distributed Video Sensor Networks*, B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, Eds. New York, NY, USA: Springer-Verlag, 2011, pp. 119–136.
[18] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, “PROST: Parallel robust online simple tracking,” in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 723–730.
[19] D. Wang, H. Lu, and M.-H. Yang, “Online object tracking with sparse prototypes,” *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 314–325, Jan. 2013.
[20] X. Ren and J. Malik, “Learning a classification model for segmentation,” in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 10–17.
[21] L. Lu and G. D. Hager, “A nonparametric treatment for location/segmentation based visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[22] X. Ren and J. Malik, “Tracking as repeated figure/ground segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[23] A. Vedaldi and S. Soatto, “Quick shift and kernel methods for mode seeking,” in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 705–718.
[24] G. Mori, X. Ren, A. A. Efros, and J. Malik, “Recovering human body configurations: Combining segmentation and recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004, pp. 326–333.
[25] S. Wang, H. Lu, F. Yang, and M.-H. Yang, “Superpixel tracking,” in *Proc. IEEE ICCV*, Nov. 2011, pp. 1323–1330.
[26] S. Avidan, “Ensemble tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 494–501.
[27] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. Dickinson, and K. Siddiqi, “Turbopixels: Fast superpixels using geometric flows,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.
[28] A. Radhakrishna, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “Slic superpixels,” Dept. School Comput. Commun. Sci., EPFL, Lausanne, Switzerland, Tech. Rep. 149300, 2010.
[29] C. Bibby and I. D. Reid, “Robust real-time visual tracking using pixel-wise posteriors,” in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2008, pp. 831–844.
[30] P. Chockalingam, S. N. Pradeep, and S. Birchfield, “Adaptive fragments-based tracking of non-rigid objects using level sets,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1530–1537.
[31] D. Tsai, M. Flagg, and J. M. Rehg, “Motion coherent tracking with multi-label MRF optimization.” in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2010, pp. 1–11.
[32] M. Godec, P. M. Roth, and H. Bischof, “Hough-based tracking of non-rigid objects,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 81–88.
[33] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[34] R. T. Collins, “Mean-shift blob tracking through scale space,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. 234–240.
[35] K. Nummiaro, E. Koller-Meier, and L. V. Gool, “An adaptive color-based particle filter,” *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, 2003.
[36] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The PASCAL visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.



Fan Yang (S’10) received the M.Sc. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2011. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Maryland, College Park, MD, USA. His research interests include computer vision and pattern recognition. He has published several papers on visual tracking and object detection. He is a recipient of the Dean’s Fellowship from the University of Maryland College Park in 2011 and 2012.

Huchuan Lu (SM’12) received the Ph.D. degree in system engineering and the M.Sc. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 1998, respectively. He joined the faculty in 1998 and currently is a Full Professor of the School of Information and Communication Engineering, DUT. His current research interests include computer vision and pattern recognition with focus on visual tracking, saliency detection, and segmentation. He is a member of the ACM and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS.

Ming-Hsuan Yang (SM’06) received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2000. He is currently an Associate Professor of electrical engineering and computer science with the Department of Electrical Engineering and Computer Science, University of California, Merced, CA, USA. He serves as a Program Chair of the Asian Conference on Computer Vision in 2014; an Area Chair for the IEEE International Conference on Computer Vision in 2011, the IEEE Conference on Computer Vision and Pattern Recognition in 2008, 2009, and 2014, the European Conference on Computer Vision in 2014, and the Asian Conference on Computer in 2009, 2010, and 2012. He has served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and currently is an Associate Editor of the *International Journal of Computer Vision, Image and Vision Computing*, and the *Journal of Artificial Intelligence Research*. He is a recipient of the National Science Foundation CAREER Award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a Senior Member of the ACM.