Superpixel Tracking

Shu Wang[1], Huchuan Lu[1], Fan Yang[1], and Ming-Hsuan Yang[2]
[1]School of Information and Communication Engineering, Dalian University of Technology, China
[2]Electrical Engineering and Computer Science, University of California at Merced, United States

Abstract

While numerous algorithms have been proposed for object tracking with demonstrated success, it remains a challenging problem for a tracker to handle large change in scale, motion, shape deformation with occlusion. One of the main reasons is the lack of effective image representation to account for appearance variation. Most 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 mid-level 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 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. 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.

1. 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, 3, 10, 8, 21, 11]. While low-level cues are effective for feature tracking and scene analysis, they are less effective in the context of object tracking [23]. On the other hand, numerous works have demonstrated that adaptive appearance models play a key role in achieving robust object tracking [9, 4, 13, 1, 11, 20].

In [13], 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 [2] 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. The Fragment-based tracker [1] aims to solve partial occlusion with a representation based on histograms of local patches. The tracking task is carried out by combing 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.

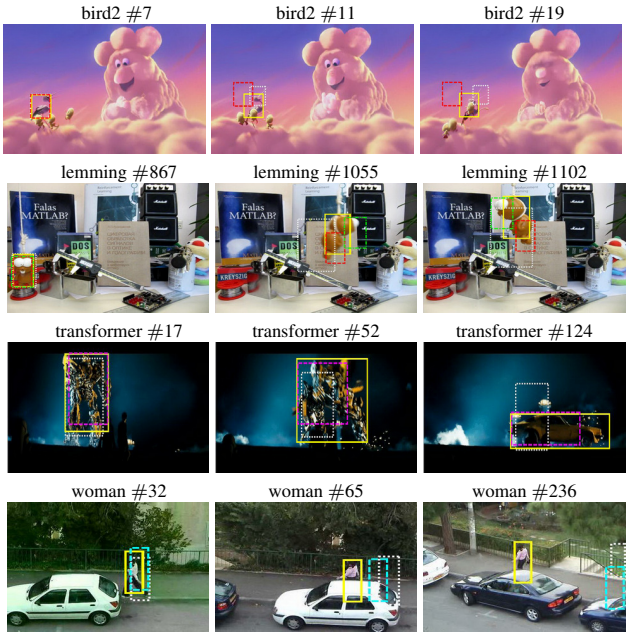


Figure 1. Four common challenges encountered in tracking. The results by our tracker, IVT [13], VTD [11], PROST [20], FragTrack [1] and PDAT [10] methods are represented by yellow, red, white, green, cyan, and magenta rectangles. Existing trackers are not able to effectively handle heavy occlusion, large variation of pose and scale, and non-rigid deformation, while our tracker gives more robust results.

In addition to account for appearance variation, recent works have focused on reducing visual drifts. In [3], an algorithm extends multiple instance learning to an online setting for object tracking. Whereas it is able to reduce visual drifts, this method is not able to handle large non-rigid shape deformation. The PROST method [20] 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. 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 [11]. 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.

Mid-level visual cues have been effective representations with sufficient information of image structure and great flexibility when compared with high-level appearance models and low-level features. In particular, superpixels have been one of the most promising representations with demonstrated success in image segmentation and object recognition [18, 15, 22, 12, 17]. 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 [19], a tracking method based on superpixel 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 CRF for region matching, the computational complexity is rather high. Further, it is not designed to handle complex scenes including heavy occlusion and cluttered background as well as large lighting change.

Similarly, a non-parametric method [14] also aims to segment one single salient foreground object from background..

In this paper, we exploit effective and efficient mid-level visual cues for object tracking with superpixels. During the training stage, the segmented superpixels are grouped for constructing a discriminative appearance 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. 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.

2. Proposed Algorithm

We present details of the proposed image representation scheme and tracking algorithm in this section.

2.1. Bayesian Tracking Formulation

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

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

where X_t is the state at time t, $Y_{1:t}$ is all the observations up to time t, and α is a normalization term. In this work, the target state is defined as $X_t = (X_t^c, X_t^s)$, where X_t^c represents the center location of the target and X_t^s denotes its scale. 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:

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

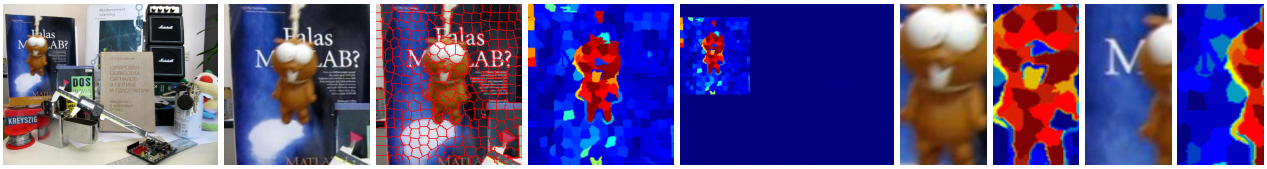
where $\hat{C}(X_t)$ represents the confidence of an observation at state X_t being the target. The state estimate of the target $\hat{X}_t$ at time t can be obtained by the MAP estimate over the N samples at each time t. Let $X_t^{(l)}$ denote the l-th sample of the state X_t,

$$\hat{X}_t = \arg\max_{X_t^{(l)}} p(X_t^{(l)}|Y_{1:t}) \ \forall\, l = 1, ..., N \quad (3)$$

In the following, the superpixel-based discriminative appearance model for tracking is introduced in Section 2.2, followed by construction of the confidence map based on this model in Section 2.3. The observation and motion models are presented in Section 2.4, and then the update scheme.

2.2. Superpixel-based Discriminative Appearance Model

To construct an appearance model for both the target and the background, prior knowledge regarding the label of each



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 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) segmentation result of (b). (d) the computed confidence map of superpixels using Eq. 7 and Eq. 8. 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.

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:

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

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

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

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[1] of the target in the t-th training frame into N_t superpixels. Each superpixel $sp(t, r)$ $(t = 1, ..., m,\ r = 1, ..., N_t)$ is represented by a feature vector f_t^r. of . Next, we apply the mean shift clustering algorithm [6] on the total feature pool $F = \{f_t^r | t = 1, ..., m;\ r = 1, ..., N_t\}$, and obtain n different clusters. In the feature space, each cluster $clst(i)$ $(i = 1, ..., n)$ is represented by its cluster center $f_c(i)$, its cluster radius $r_c(i)$ and its own cluster members $\{f_t^r | f_t^r \in clst(i)\}$.

Now that every $clst(i)$ corresponds to its own image region $S(i)$ in the training frames (image regions that superpixel members of $clst(i)$ cover), we count two scores for each $clst(i)$: $S^+(i)$ 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 greater the ratio $S^+(i)/S^-(i)$ is, the more likely superpixel members of $clst(i)$ appear in target area in training frames. Consequentially, we give each cluster a target-background confidence measure between 1 and -1 to indicate how probable its superpixel members belong to the target or background:

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

Our superpixel-based discriminative appearance model is constructed based on four factors: cluster confidences C_i^c, cluster centers $f_c(i)$, 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. By applying the confidence measures of each cluster to superpixels in the training frames, we are able to learn a similar prior knowledge as Eq. 5 from a set of training images.

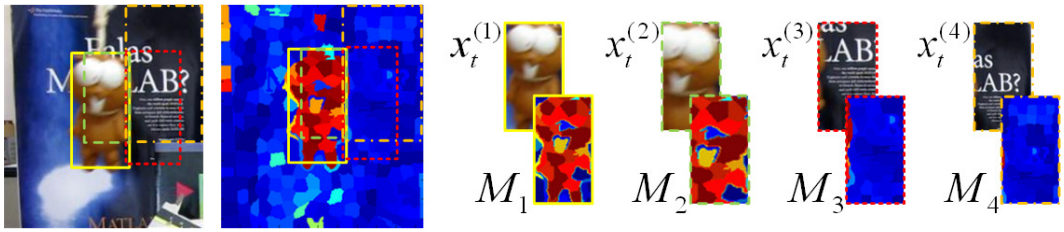
The merits of the proposed superpixel-based discriminative appearance model are shown by Figure 4 and Section 3: Few background superpixels appearing in the target area (as a result of drifts or occlusions), are likely to be clustered into the same group with other background superpixels, and thus have negligible effect to our algorithm during training and update.

2.3. 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 Figure 2 (b) and (c)). To compute a confidence map for current frame, we evaluate every superpixel and compute its confidence measure. The confidence measure 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 $clst(i)$ in the feature space, then the target-background confidence of

[1] 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. The parameter λ_s is a stable parameter, which controls the size of this surrounding region, and is set to 1.5 in all experiments.

[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

Figure 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 2.4.

$clst(i)$ 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 $clst(i)$. The confidence measure of each superpixel is computed as follows:

$$w(r,i) = \exp(-\lambda_d \times \frac{||f_t^r - f_c(i)||_2}{r_c(i)}) \quad \forall\, r = 1, ..., N_t, \;\; i = 1, ..., n \tag{7}$$

$$C_r^s = w(r,i) \times C_i^c, \;\forall\, r = 1, ..., N_t \tag{8}$$

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 $clst(i)$ 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 measure 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. We assign every pixel in the superpixel $sp(t,r)$ with superpixel confidence C_r^s, and every pixel outside this surrounding region with -1. Figure 2 (a)-(e) shows 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 2.2. In turn, the following steps for identifying the likely locations of the target in object tracking are based on this confidence map.

2.4. 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 (See details in the supplemental material).

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

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

However, this target-background confidence C_l does not deal with scaling well. In order to make the tracker robust to the scaling of the target, we weigh C_l with respect to the size of each candidate as follows:

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

where $S(X_t)$ represents the area size of target state X_t. 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 $p(Y_t|X_t^s)$ adaptive to scale. Figure 3 illustrates this weighting scheme. We then 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:

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

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 estimate $\hat{X}_t$ can be computed with Eq. 3. Figure 2 (f)-(i) show two drawn samples and their corresponding confidence maps. As shown in

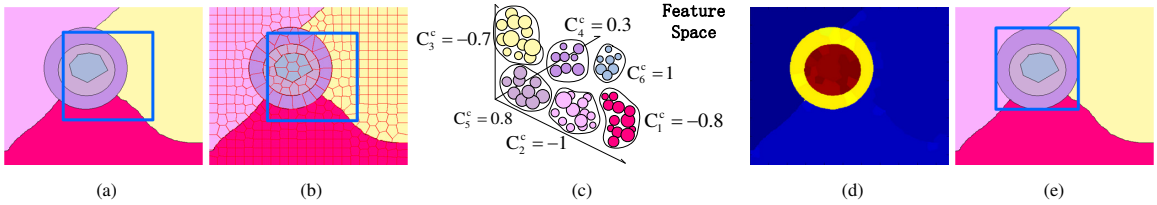


Figure 4. Recovering from drifts. (a) a target object with visual drifts. (b) the surrounding region of the target is segmented into superpixels. (c) clustering results of (b) in feature space and the target-background confidence of each cluster. (d) the confidence map in a new frame computed with clustering results. (e) the MAP estimate of the target area (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 our tracker with a more discriminative power against drifts than holistic appearance models.

these examples, the confidence maps facilitate the process of determining the most likely target location.

2.5. 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. A sliding window update scheme is adopted, in which a sequence of H frames is stored during tracking process. For every U frames, we put a new frame into this sequence, and delete the oldest one. That is, this process retains a record from 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[3], and this process is the same as the training process described in Section 2.2.

With the proposed discriminative appearance model using mid-level cues, we present a simple but efficient method to handle occlusion in object tracking. 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 highest confidence of belonging to the target, and conversely the lower bound indicates all pixels belonging to the background. We set a threshold θ_o to detect heavy or full occlusions:

$$\frac{\mu_C - \max(\{C_l\}_{l=1}^N)}{S(X_t^{(l)}) \times 2} > \theta_o \qquad (13)$$

where μ_C is the average of confidence (from Eq. 10) of the target estimates in the retained sequence of H frames. The numerator of the left hand side of this formula reflects the difference between the confidence C_l of the MAP estimate of current frame, and the average confidence of target in the retained sequence. The denominator is a normalizing term to confine the left hand side to the range of [-1,1]. If the confidence C_l of the MAP estimate of current frame is much less than the average of confidence of the retained sequence, that means that the MAP target estimate is still of high probability to be background area, then Eq. 13 is satisfied and a heavy occlusion is deemed to occur. In such situations, the target estimate X_{t-1} of the last frame is considered the target estimate $\hat{X}_t$ for the current frame. Furthermore, instead of deleting the oldest (first) 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 will not delete all information of target when long time heavy occlusion occurs, and keep on learning the occlusion at the same time. Without learning the appearance of heavy occlusion, our tracker may be affected when occlusion changes. All superpixels in the current frame are regarded as lying in the background area and μ_C is saved as the confidence of current frame. As will be shown in the experiments, robust results can be obtained with this scheme.

[3]The length of information sequence H and the spacing interval U is set 10 and 3 in our experiments. The update frequency W is set between 5 and 10.

Table 1. Proposed algorithm.

Initialization:

for $t = 1$ to m (e.g., m is set to 4 in all experiments)

1. Initialize parameters of our algorithm in the first frame.
2. Segment the surrounding region of X_t for training into N_t superpixels, and extract their features $\{f_t^r\}_{r=1}^{N_t}$.

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-1} into N_t superpixels and extract their features. Compute the target-background confidence map using Eq. 7 and Eq. 8.
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 Eq. 10-12.
4. Estimate MAP state $\hat{X}_t$ using Eq. 3.
5. Detect 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

The confidence map with update is also used to recover our tracker from drifts. Figure 4 illustrates how the proposed method recovers from drifts with the information from superpixels and confidence map. The main steps of the proposed algorithm are summarized in Table 1.

3. Experimental Results

We present the experimental setups and empirical results as well as observations in this section.

3.1. Experimental Setups

We utilize normalized histogram in the HSI color space as the feature for each superpixel. The SLIC algorithm [17] is applied to segment frames into superpixels where the spatial proximity weight and number of superpixels are set to 10 and 300, respectively. The bandwidth of the mean shift clustering [6] is set to the range of 0.15 and 0.20. We note that the bandwidth needs to be wide enough to separate superpixels from the target and background into different clusters. 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 σ_c and σ_s in Eq. 9 are set between 3 and 8 in anticipation of the fastest motion speed or changing scale of the target objects. The threshold to detect occlusion θ_o is between the range of 0.1 and 0.3.

We evaluate our algorithm on 10 challenging sequences (6 from prior work [1, 10, 20, 11] and 4 from 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 Figure 1, Figure 5 and Figure 6). The quantitative evaluations of the Mean Shift (MS) [5], adaptive color-based particle filter (PF) [16], IVT [13], FragTrack [1], MILTrack [3], PROST [20], VTD [11] methods and our tracker are presented in Table 2, Table 3 and Figure 7. More results and videos can be found in the supplementary material and at our web site (`http://faculty.ucmerced.edu/mhyang/pubs/iccv11a.html`). In addition, our work can easily be extended to segment salient foreground target from background, and results are presented in supplemental material. All the MATLAB code and datasets are available on our web site.

3.2. Empirical Results

We first evaluate our algorithm with the sequences used in prior works: *singer1* and *basketball* from VTD [11], *transformer* from PDAT [10], *lemming* and *liquor* from PROST [20], and *woman* from FragTrack [1]. We then test 4 sequences from our own dataset: *bolt*, *bird1*, *bird2*, and *girl*. For fair comparison, we carefully adjust the parameters of every tracker with the code provided by the authors and use the best result from 5 runs, or taken directly from the presented results in the prior works.

Sequence	MS	PF	IVT	Frag	MIL	PROST	VTD	SPT
lemming	236	184	**14**	84	**14**	23	98	**7**
liquor	137	28	296	31	165	**22**	155	**9**
singer1	116	25	5	21	20	–	**3**	**4**
basketball	203	21	120	14	104	–	**11**	**6**
woman	**32**	79	133	112	120	–	109	**9**
transformer	46	49	131	47	**33**	–	43	**13**
bolt	204	34	386	100	380	–	**14**	**6**
bird1	330	**137**	230	228	270	–	251	**15**
bird2	73	75	115	24	**13**	–	46	**11**
girl	304	**16**	184	106	55	–	57	**21**

Table 2. Tracking results. The numbers denote average errors of center location in pixels.

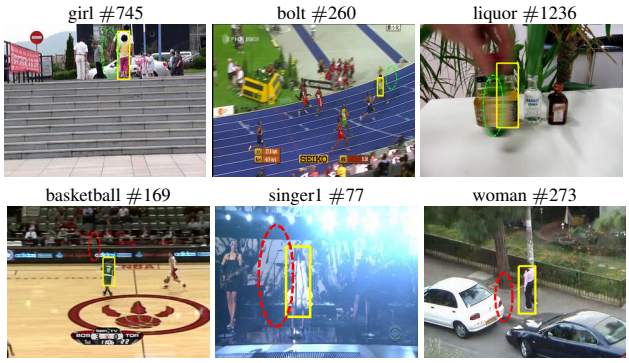


Figure 5. Tracking results with comparisons to color-based trackers. The results by the MS tracker, PF method and our algorithm are represented by red ellipse, green ellipse and yellow rectangles. It is evident that our tracker is able to handle cluttered background (*girl* and *basketball* sequences), drastic movement (*bolt* sequence), heavy occlusion (*liquor* and *woman* sequences) and lighting condition change (*singer1* sequence).

Comparison with color-based trackers:

As shown in Figure 5, the adaptive color-based particle filter [16] can neither deal with cluttered background, drastic movement nor heavy occlusion. The mean shift tracker with adaptive scale [5] does not perform well when there is a large appearance change due to non-rigid motion, lighting change and heavy occlusion (Figure 5). We note that this tracker is designed to handle scale change. However, it is less effective in dealing with lighting and occlusion.

On the other hand, the discriminative appearance model based on mid-level representation alleviates negative influences from noise and background clutter. Consequently, our tracker is able to track objects undergoing heavy occlusion, non-rigid deformation and lighting change in clutter backgrounds (Figure 5).

Comparison with other state-of-the-art trackers:

Visual 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

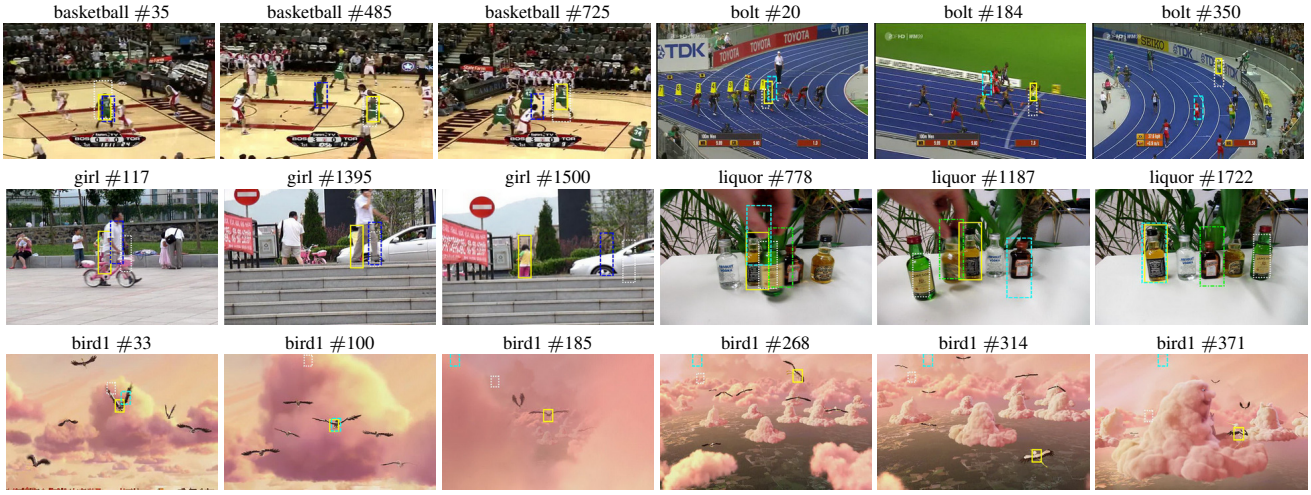


Figure 6. Tracking results. The results by our tracker, IVT, VTD, PROST, MILTrack and FragTrack methods are represented by yellow, red, white, green, blue and cyan rectangles.

Sequence	MS	PF	IVT	Frag	MIL	PROST	VTD	SPT
lemming	171	426	1046	678	**1105**	969	471	**1290**
liquor	413	1202	380	1375	353	**1444**	471	**1701**
singer1	64	96	**332**	87	87	–	**350**	297
basketball	78	455	80	512	204	–	**601**	**707**
woman	35	31	**49**	44	38	–	27	**310**
transformer	28	32	29	38	30	–	**47**	**124**
bolt	15	172	4	32	12	–	**195**	**224**
bird1	1	6	4	47	**114**	–	7	**139**
bird2	36	19	9	42	**86**	–	9	**94**
girl	79	**1106**	107	628	560	–	828	**1180**

Table 3. Tracking results. The numbers denote the count of successful frame based on evaluation metric of the PASCAL VOC object detection [7] which is also used in other tracking algorithm [20]. Note that we use elliptical target area for the mean shift tracker (MS) and the adaptive color-based particle filter (PF) to calculate the metric used in PASCAL VOC tests for fair comparison.

that these trackers typically focus on learning target appearance rather than the background (i.e., with a generative approach). As shown in the first row (*bird2* sequence) of Figure 1, the IVT and VTD methods drift away from the target into background regions when heavy occlusions occur in frame 11 and 19.

In the *basketball* and *bolt* sequences (shown in Figure 6), the IVT, MILTrack and FragTrack methods drift to background area in early frames for that they are not designed for non-rigid deformation. Although the VTD tracker achieves the second best results in these two sequences, 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 imprecise tracking results. In contrast, the discriminative appearance model of our tracker utilizes background information effectively and avoids such drifting problems throughout these two sequences.

Large variation of pose and scale: The second row (*lemming* sequence) in Figure 1 shows that, the IVT, MILTrack and PROST methods perform well as the methods with holistic appearance models are effective for tracking rigid targets (one tracker in PROST is an off-line template). They are able to track the target well when there is no large change in scale and pose (e.g., out-of-plane rotation). However, it is not surprising that their holistic appearance models (where target objects are enclosed with rectangles for representation) are not effective in accounting for appearance change due to large pose change. On the other hand, our tracker is more robust to pose variation due to the use of mid-level appearance model, and outperforms other trackers as the proposed superpixel-based discriminative appearance model learns the difference between the target and background with updates, which makes our tracker able to handle scaling and occlusion throughout this sequence.

Large shape deformation: The third row (*transformer* sequence) of Figure 1 shows one example when drastic shape deformation occurs, tracking algorithms using holistic appearance models or blobs are unlikely to perform well (IVT, MIL and VTD). The patch-based dynamic appearance tracker (PDAT) [10] is able to track the target object in this sequence as its representation scheme is based on local patches and not sensitive to non-rigid shape deformation. Nevertheless, without sufficient usage of the appearance information of both target and background, the tracking results are less accurate. Our appearance model utilizes information of both target and background on local mid-level cues, and distinguishes target parts from background blocks precisely. Thus our tracker gives the most accurate results.

Heavy occlusion: The target in the *liquor* sequence undergoes heavy occlusion for many times (the second row of Figure 6). Since our superpixel-based discriminative ap-

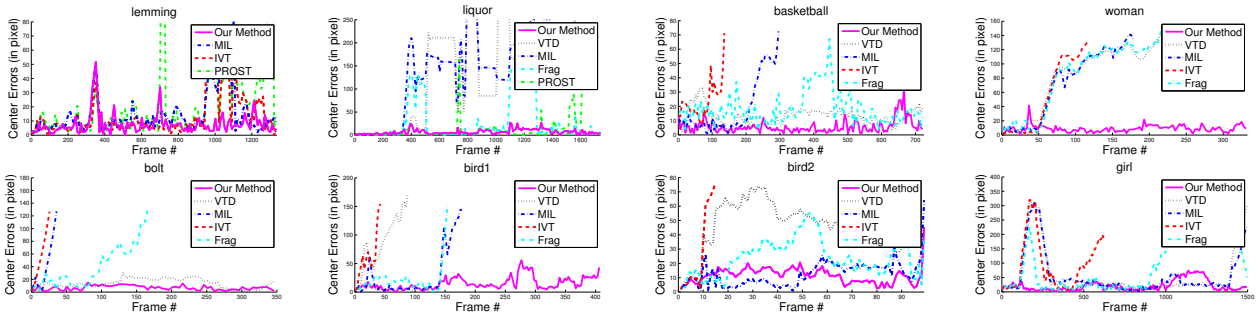


Figure 7. Tracking results comparison of IVT, Visual Tracking Decomposition (VTD), MILTrack, FragTrack, PROST and our tracker.

pearance model is able to alleviate influence from background pixels and learns the appearance of both target and background with superpixels, our tracker is able to detect and handle all heavy occlusions accordingly. Although the PROST method may recover from drifts after occlusion, it does not succeed all the time. Furthermore, the other trackers fail for that they are not able to handle large appearance change due to heavy occlusion or recover from drifts.

In the *bird1* sequence (third row in Figure 6), the target object undergoes significant non-rigid deformation, rapid motion, pose change, and occlusion for a long duration. Unless a tacker is able to distinguish foreground from background based on low-level or mid-level cues, it is unlikely to handle heavy occlusion and non-rigid deformation simultaneously. Our discriminative appearance model with superpixels enables our tracker to detect full occlusion and account for shape deformation at the same time. The other trackers fail mainly due to large appearance change caused by heavy occlusion and shape deformation.

In addition to the above-mentioned results, our tracker outperforms other state-of-the-art methods in dealing with heavy occlusion in the *woman* and *girl* sequences (shown in Figure 1 and Figure 6).

4. Conclusion

In this paper, we propose a robust tracker based on a discriminative appearance model and superpixels. We show that the use of superpixels provide flexible and effective mid-level cues, which are incorporated in an appearance model to distinguish the foreground target and the background. 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 proposed tracker performs favorably against existing state-of-the-art algorithms in the literature.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[2] S. Avidan. Ensemble tracking. In *CVPR*, pages 494–501, 2005.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[4] R. Collins and Y. Liu. On-line selection of discriminative tracking features. In *ICCV*, pages 346–352, 2003.

[5] R. T. Collins. Mean-shift blob tracking through scale space. In *CVPR (2)*, pages 234–240, 2003.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[7] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[8] B. Han, Y. Zhu, D. Comaniciu, and L. S. Davis. Visual tracking by continuous density propagation in sequential Bayesian filtering framework. *PAMI*, 31(5):919–930, 2009.

[9] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. In *CVPR*, pages 415–422, 2001.

[10] 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 *CVPR*, pages 1208–1215, 2009.

[11] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

[12] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *PAMI*, 31(12):2290–2297, 2009.

[13] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *NIPS*, pages 793–800. MIT Press, 2005.

[14] L. Lu and G. D. Hager. A nonparametric treatment for location/segmentation based visual tracking. In *CVPR*, 2007.

[15] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, pages 326–333, 2004.

[16] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool. An adaptive color-based particle filter. *Image Vision Comput.*, 21(1):99–110, 2003.

[17] A. Radhakrishna, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. Technical Report 149300, EPFL, 2010.

[18] X. Ren and J. Malik. Learning a classification model for segmentation. In *ICCV*, pages 10–17, 2003.

[19] X. Ren and J. Malik. Tracking as repeated figure/ground segmentation. In *CVPR*, 2007.

[20] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *CVPR*, pages 723–730, 2010.

[21] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli. SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors. In *CVPR*, pages 2937–2944, 2009.

[22] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *ECCV*, pages 705–718, 2008.

[23] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):1–45, 2006.