Structured Visual Tracking with Dynamic Graph

Zhaowei Cai[1], Longyin Wen[1], Jianwei Yang[1], Zhen Lei[1], and Stan Z. Li[1,2,⋆]

[1]CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences
[2]China Research and Development Center for Internet of Thing
{zwcai,lywen,jwyang,zlei,szli}@cbsr.ia.ac.cn

Abstract. Structure information has been increasingly incorporated into computer vision field, whereas only a few tracking methods have employed the inner structure of the target. In this paper, we introduce a dynamic graph with pairwise Markov property to model the structure information between the inner parts of the target. The target tracking is viewed as tracking a dynamic undirected graph whose nodes are the target parts and edges are the interactions between parts. These target parts within the graph waiting for matching are separated from the background with graph cut, and a spectral matching technique is exploited to accomplish the graph tracking. With the help of an intuitive updating mechanism, our dynamic graph can robustly adapt to the variations of target structure. Experimental results demonstrate that our structured tracker outperforms several state-of-the-art trackers in occlusion and structure deformations.

1 Introduction

Visual tracking is an important area in computer vision community, and it has a number of applications in the areas of video surveillance, human-computer interaction, behavior analysis, etc. Generally speaking, most of recent tracking methods focus on three aspects to improve the accuracy and robustness: feature, such as pixel values [1], color [2], and texture [3, 4], representation model, such as subspace learning [1], SVM [5], Boosting [3, 4] and sparse representation [6, 7], and structure information, including [8–10]. Although structure information has been widely considered in the fields of object detection [11], object recognition [12], etc, only a few trackers take it into account.

In this paper, we introduce a structure model to improve the robustness of our tracker to structure deformation. The structure information is generated by oversegmenting the target into several parts (superpixels) and modeling the interactions between the neighboring parts. The appearances of parts and their relations are incorporated into a dynamic undirected graph with pairwise Markov property. Therefore, the tracking problem in our method is viewed as the tracking of the undirected graph, which is also a matching problem between the target graph $\mathcal{G}(V', E')$ and the candidate graph $\mathcal{G}(V, E)$. During the process of tracking, the candidate target parts are cut out with the help of MRF spatial

⋆ Corresponding author

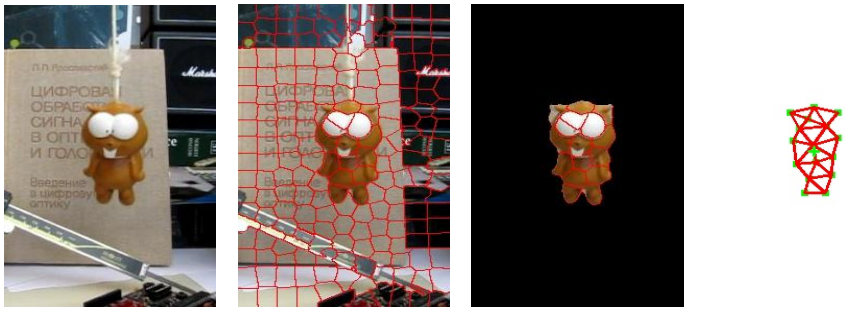
prior, which is optimized by Graph Cut [13]. At the step of graph matching, the optimal matching from candidate graph to target graph is interpreted as finding the main cluster from assignment graph with spectral technique. The final position of the target is determined by a series of successfully matched parts based on their appearance likelihood and the relative location displacement with the neighboring parts which is called structure likelihood.

The contributions of this work are summarized as follows. Firstly, the structure information is considered throughout our tracking process including candidate target parts selection, graph matching and target center location. Secondly, an efficient moving object segmentation method at the level of superpixel is developed. Another contribution is that we firstly introduce graph matching into tracking task. Finally, an intuitive and effective updating mechanism of dynamic target graph is proposed to adapt to the structure variations of the target.

2 Related Works

General tracking approaches [14, 4, 3] represent the target as a bounding box template, and intuitively no structure information is considered. An online incremental subspace is modeled in [1] to robustly represent the target, which obtains good performance with illumination variations. However, the rigid template updating strategy undermines its robustness to non-rigid distortion and occlusion. Some other trackers [3–5] employ SVM or Boosting classifiers to model the difference between the target and the background. Similarly, the ignorance of structure information also results in the bad performance in structure deformation and occlusion. [6, 7] model the target as a sparse representation of a dictionary constructed with historical parts appearance features, which are insensitive to occlusion, whereas they fail to effectively adapt to structure variations.

The part-based model has wide applications in object recognition [12] and detection [11], in which the states of parts are optimized as hidden states within Conditional Random Field (CRF) [12] and Latent Support Vector Machine (LSVM) [11] respectively. This kind of method needs a large number of training samples to find the optimal structure model, which is infeasible in visual tracking due to inadequate training samples and high computation complexity. There is also a few part-based models arising in tracking field [15, 9, 8, 2, 10]. Although the target is represented as manually labeled parts in [15], only limited structure information is incorporated and the template of each part does not update ever. In [9], the fixed number of parts are generated without updating and the relative locations of parts are fixed too. Kwon et al. [8] take SIFT descriptor as its part generation method which is very unstable and the tracking results usually include bad tracked parts. The trackers proposed in [2, 10] rely on oversegmentation to generate parts. The former tracker [2] only computes the probabilities of parts belonging to the foreground without any structure constraints, which is easily drifted away by other color-similar objects and whose tracking results will shrink to unoccluded parts when occlusion happens. The latter tracker [10] models the low-level superpixel correspondence with CRF during the process



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

Fig. 1. (a) is the tracking window, (b) is the superpixel oversegmentation result, (c) is the foreground/background separation result, and (d) is the construction of our undirected graph with Markov property, where green squares represent nodes and red lines mean the interactions between neighboring nodes.

of figure/ground segmentation, whereas the high complexity of CRF limits its further application in visual tracking task.

Another work similar to ours is [16], in which Yang et al. resort to data mining technique to discover auxiliary objects and combine them with the target into a star topology graphical model for robust visual tracking. Its difference from ours is that we focus on the structured representation of the target instead of the correlation between the target and the surroundings. Our work also relates to motion segmentation [17] and video segmentation [18], because the separation of candidate target parts from background needs segmentation. However, our segmentation is exploited to collect the candidate target parts, hence rough segmentation at the level of superpixel instead of pixel is enough here.

3 Structured Appearance Model

At the beginning of our tracking, the tracking window is segmented into a set of parts. To better represent the target with compact and perceptually meaningful parts, we group pixels into superpixels. The *Simple Linear Iterative Clustering* (SLIC) algorithm proposed in [19] is adopted here because it is efficient and has compact results, and the results are shown in Fig. 1(b).

3.1 Spatial Prior with Graph Cut

Given a set of superpixels $\{T_p\}$, we need to collect the candidate target parts set $\{T_i\}^P$ and construct the candidate graph $\mathcal{G}(V, E)$. A pairwise Markov Random Field (MRF) is built here to separate the candidate foreground parts from background. To better and more robustly represent the appearance of the target, a generative target color histogram and a discriminative SVM classifier are

simultaneously combined to compute the unary potentials. The MRF energy is specifically optimized as:

$$E(\mathbf{L}) = \sum_{p \in S} D_p^g(l_p) + \alpha_1 \sum_{p \in S} D_p^d(l_p) + \alpha_2 \sum_{p,q \in N} V_{p,q}(l_p, l_q) \tag{1}$$

where $\mathbf{L} = \{l_p | l_p \in \{0, 1\}, p \in S\}$ is the labeling of superpixels set $\{T_p\}$. $D_p^g(l_p)$ and $D_p^d(l_p)$ represent the unary potentials for superpixel T_p gotten from color histogram and SVM classifier respectively. $V_{p,q}(l_p, l_q)$ indicates the pairwise potential for interacting superpixels T_p and T_q. S is the set of superpixels in the tracking window, and N is the set of interacting pairs of superpixels who have geometrically adjacent edges (red lines in Fig. 1(b)). α_1 and α_2 are the constant parameters that balance the influences of these potential terms. We decide to use Graph Cut [13] to solve the minimization of Equ. 1 because of its high running efficiency. Therefore, the summation of $D_p^g(l_p)$ and $D_p^d(l_p)$ in Equ. 1 can be viewed as the T-link energy, and $V_{p,q}(l_p, l_q)$ as the N-link energy in the maxflow framework [13], where T-links connect superpixels with terminals and N-links connect pairs of neighboring superpixels.

For the generative model, the normalized target RGB color histograms of foreground $\mathcal{H}^f$ and background $\mathcal{H}^b$ are applied, and then the potential of every superpixel is derived from the probability of every pixel $P(C_i|\mathcal{H}^b)$ within it.

$$D_p^g(l_p) = \begin{cases} -\frac{1}{N_p} \sum_{i=1}^{N_p} \log P(C_i|\mathcal{H}^b) & l_p = 0 \\ -\frac{1}{N_p} \sum_{i=1}^{N_p} \log P(C_i|\mathcal{H}^f) & l_p = 1 \end{cases} \tag{2}$$

where C_i is the RGB value of pixel i, and N_p is the number of pixels in the superpixel T_p. For the discriminative model, an online SVM [20] classifier is applied and it is trained with RGB color histograms of superpixels.

$$D_p^d(l_p) = \begin{cases} \lambda_1 \widehat{y}(f_p) & l_p = 0 \\ 1 - \lambda_2 \widehat{y}(f_p) & l_p = 1 \end{cases} \tag{3}$$

where $\widehat{y}(f_p) = \mathbf{w} \cdot \Phi(f_p) + b$ is the discriminant function of SVM classifier, f_p is the color histogram of superpixel T_p, and λ_1 and λ_2 are the constants. The pairwise term $V_{p,q}(l_p, l_q)$ captures the discontinuity between two neighboring superpixels.

$$V_{p,q}(l_p, l_q) = \exp\{-D(f_p, f_q)\} \tag{4}$$

where $D(f_p, f_q)$ is the $\mathcal{X}^2$ distance between the color histograms f_p and f_q. With the help of the foreground/background segmentation, the candidate graph $\mathcal{G}(V, E)$ is constructed whose nodes are the cut out candidate parts and edges are the interactions between nodes, as shown in Fig. 1(d). We define two parts have interaction or relation if their location distance is smaller than $\theta_d \cdot r$, where θ_d is the constant. $r = \sqrt{w \cdot h / N_s}$ is the average radius of superpixels, where w and h are the width and height of the tracking window respectively and N_s is the number of superpixels in the tracking window.

3.2 Graph Matching with Spectral Technique

In this section, we firstly introduce graph matching into the tracking field and explain its importance. A key aspect for our part-based tracker is to find the correspondence between parts in sequential frames, and the correspondence is completed with the help of the target inner structure beside the appearance of inner parts. Therefore, the graph matching problem can be viewed as an optimization problem, in which the appearance features of parts as well as the geometric constraints between parts are incorporated. Given a set of M_p candidate parts $\{T_i\}^P$ in $\mathcal{G}(V, E)$ which are separated from current frame, and a set of M_q target parts $\{T_{i'}\}^Q$ in $\mathcal{G}(V', E')$which are collected from historical frames, a correspondence mapping of assignments is a binary value set $\mathbf{x} = (x_1^1, \cdots, x_{m_p}^k, \cdots, x_{M_p}^K)$ where $x_{m_p}^k \in \{0, 1\}$, $k = 1, 2, \cdots, K$, $m_p = 1, 2, \cdots, M_P$, and K is the number of candidate assignments for single part T_i which is chosen based on feature similarity. For simplicity, we use x_a to represent $x_{m_p}^k$ in $\mathbf{x}$. For each assignment $a = (i, i')$, the appearance features similarity d_a between T_i and $T_{i'}$ is applied to indicate how well these two parts are matched. For each pair of assignments (a, b), where $b = (j, j')$, an affinity measure $\theta_{a,b}$ is used to evaluate the compatibility between assignments a and b, that is the geometric affinity between the parts pair (i, j) in $\mathcal{G}(V, E)$ and the parts pair (i', j') in $\mathcal{G}(V', E')$. Therefore, the set of assignments can be incorporated into an undirected assignment graph whose node weight is d_a and edge weight is $\theta_{a,b}$, and the optimization is formulated as:

$$E(\mathbf{x}) = \sum_{a \in S} d_a x_a^2 + \sum_{a,b \in N} \theta_{a,b} x_a x_b \tag{5}$$

where S is the set of assignments and N is the set of compatible pairs of assignments. Here, we define a and b are compatible assignments if T_i and T_j are geometric neighbors, and $T_{i'}$ and $T_{j'}$ are geometric neighbors at the same time. Only one-to-one matching is allowed in our model, hence Equ. 5 has to subject to the mapping constraints:

$$\sum_{i=m_p, i'} x_a \leq 1, \sum_{i, i'=m_q} x_a \leq 1 \tag{6}$$

We transfer the assignment graph into an affinity symmetry matrix M, where

- $M(a, a) = d_a = \exp\{-D(f_i, f_{i'})\}$ indicates how well an individual assignment is matched, where $D(f_i, f_{i'})$ is the $\mathcal{X}^2$ distance between two color features f_i and $f_{i'}$ of parts T_i and $T_{i'}$ respectively.
- $M(a, b) = \theta_{a,b}$ denotes how well the two geometrically relative pairwise assignments a and b are compatible. If the two assignments are not compatible, we set $\theta_{a,b} = 0$, otherwise, $\theta_{a,b} = \beta \cdot \exp\{-\frac{1}{r}\|v_{i,j} - v_{i',j'}\|_2\}$, where $v_{i,j}$ is the location vector from part T_i to T_j, β represents the influence of the pairwise term and r is the average radius of superpixels. The property of undirected graph owns the symmetry $M(a, b) = M(b, a)$.

Then, Equ. 5 can be reformulated as $E(\mathbf{x}) = \mathbf{x}M\mathbf{x}^T$. Given the mapping constraints in Equ. 6, the optimal assignment solution is $\mathbf{x}^* = \arg\max_{\mathbf{x}} (\mathbf{x}M\mathbf{x}^T)$. We prefer to apply the spectral approach to solve the above objective function. Similar to [21], graph matching equals to find the main cluster from the assignment graph and the optimization can be solved by eigenvector technique in our paper. After the eigenvalue decomposition of affinity matrix M, the values of main eigenvector are interpreted as the confidence of corresponding assignments. We greedily and sequentially accept the assignments with largest confidence and reject those assignments which are conflicted with the accepted assignments, subjected to the one-to-one matching constraint described in Equ. 6.

4 Tracking Formulation

In our tracking method, a target is represented as a dynamic undirected graph $\mathcal{G}(V', E')$, as shown in Fig. 1(d). The target state and observation at time t are defined as $Z_t = (Z_t^1, Z_t^2, \cdots, Z_t^m)$ and O_t respectively, where m is the number of parts. The state of each part T_i is $Z_t^i = (l_t^i, \Delta l_t^i)$, where l_t^i is the position of T_i, and Δl_t^i represents its location offset vector from target center. The Markov property is considered to locate the target center in our method. Therefore, the combined likelihood of each successfully matched part T_i including appearance likelihood P_a and structure likelihood P_s can be computed as:

$$\begin{aligned} P(O_t|Z_t^i) &= P_a(O_t|Z_t^i) \cdot P_s(O_t|Z_t^i) \\ &= \exp\{-D(f_i, f_{i'}) - \sum_{\substack{i,j\in N \\ i',j'\in N'}} \frac{1}{r}\|v_{i,j} - v_{i',j'}\|_2\} \end{aligned} \tag{7}$$

where $D(,)$ is the $\mathcal{X}^2$ distance between color features, N and N' are the sets of interacting parts in $\{T_i\}^P$ and $\{T_{i'}\}^Q$, $v_{i,j}$ is the geometric location vector from part T_i to T_j. The center of the target can be robustly estimated as:

$$l_c = \sum_{i=1}^{n} \frac{P(O_t|Z_t^i)}{\sum_{i=1}^{n} P(O_t|Z_t^i)}(l_t^i + \Delta l_t^i) \tag{8}$$

where n is the number of successfully matched parts, and Δl_t^i in Z_t^i is taken from its corresponding matched part directly. In order to obtain more precise target center, we perturb the target center with μ and modify the scale s so that the bounding box will cover the most positive pixels and the least negative pixels. The optimal scale s^* and perturbation μ^* can be obtained:

$$(\mu^*, s^*) = \arg\max_{\mu,s} \{\gamma \cdot N^{pos}(l_c + \mu, s) - N^{neg}(l_c + \mu, s)\} \tag{9}$$

where $N^{pos}(l_c + \mu, s)$ and $N^{neg}(l_c + \mu, s)$ are the numbers of positive pixels and negative pixels in the bounding box located at $l_c + \mu$ with scale s respectively, and γ is the term to balance the proposition of positive and negative pixels. Then the final target locates at $l_c^* = l_c + \mu^*$ with scale s^*.

Algorithm 1 Proposed Tracking Algorithm

Initialization:

1. Initialize the SVM classifier and the target color histograms $\mathcal{H}^f$ and $\mathcal{H}^b$.
2. Construct the target graph $\mathcal{G}(V', E')$ with Markov property.

Tracking:

while run **do**

1. Oversegment the tracking window into a set of superpixels $\{T_p\}$.
2. Separate the candidate target parts $\{T_i\}^P$ from the background with energy minimization in Equ. 1, and construct the candidate graph $\mathcal{G}(V, E)$.
3. Match the two graph $\mathcal{G}(V', E')$ and $\mathcal{G}(V, E)$ with the spectral technique.
4. Compute the combined likelihood of each successfully matched part $P(O_t|Z_t^i)$, and locate the optimal target center with optimal scale by Equ. 8 and Equ. 9.
5. Update the appearance model and the dynamic target graph $\mathcal{G}(V', E')$ when the updating conditions are satisfied.

end while

5 Online Update

There are two parts of our tracking framework needed to be updated online, that is the appearance model including discriminative SVM classifier and generative target color histogram, and the structure model. We adopt an online learned SVM algorithm [20] to train the discriminative classifier with the color features of newly coming samples (superpixels), which are collected periodically. Those samples which are labeled as positive by graph cut in the target bounding box are trained as positive and the others are viewed as negative. In order to avoid the drift problem caused by bad updating samples, we simultaneously add the samples collected from the initial frame into the training pool with the samples collected from current frame. For the generative foreground and background RGB histograms, we also incrementally update them, that is $\mathcal{H}_{new} = \mathcal{H}_{init} + \mathcal{H}_{old} + \mathcal{H}_{current}$, where $\mathcal{H}_{init}$, $\mathcal{H}_{old}$ and $\mathcal{H}_{current}$ are the histograms of the initial sample, the sum of historically collected samples and the newly coming sample at current frame respectively. This appearance updating mechanism not only keeps the initial information but also adapt to the appearance variations.

For updating our dynamic graph, an intuitive and effective strategy is adopted here. We define three states for nodes updating: **birth**, **stay** and **death**.

- **birth** In the final located bounding box of the target, the newly generated node i that has not been matched with any node in the target graph $\mathcal{G}(V', E')$, will be viewed at the state of birth if its geometric Euclidian distance with other node $d_{i,i'} > \theta_b \cdot r$, $\forall i' \in \mathcal{G}(V', E')$. This distance constraint prevents the newly updated $\mathcal{G}(V', E')$ from being too dense.
- **stay** The successfully matched node is viewed at the state of stay, and we define node i is successfully matched if its appearance likelihood $P_a > \theta_a$ and its structure likelihood $P_s > \theta_s$.
- **death** The node will be viewed at the state of death if it has not been successfully matched continuously for more than N_f frames.

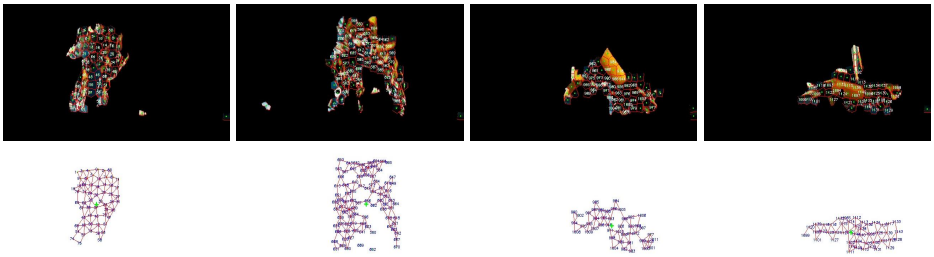


Fig. 2. The top row is the results of foreground/background separation in which the numbers on the parts mean the indices of matched target parts. The bottom row is the updating results of our dynamic graph where the numbers are the indices of target parts and the red line represents interaction between two neighboring parts.

After locating the bounding box of the target, we will delete the nodes at the state of death, keep the nodes at the state of stay and add the nodes at the state of birth into the target graph $\mathcal{G}(V', E')$, and then new edges will be constructed according to the geometric relation between nodes. This intuitive updating mechanism robustly adapts to structure variation, as illustrated in Fig. 2. The detailed process of our tracker is described in Algorithm 1.

6 Experimental Results

6.1 Experiment Setup

In order to evaluate the superiority of our tracker, we test it on 8 challenging sequences (5 of them are from prior works [22, 8, 18], and the last 3 are our own). These sequences include most of the challenges: complex environment, large scale changes, inner structure deformation, abrupt movement and severe occlusion. The quantitative comparisons of bounding box based trackers including IVT [1], MIL [4], $\ell 1$ [6], and TLD [14], and part based trackers including Frag [9], HABT [15], BHMC [8] and SPT [2], are presented in Table 1, Table 2 and Fig. 3. More tracking results and our original datasets are available at http://www.cbsr.ia.ac.cn/users/lywen/.

Our tracker is implemented in C++ code and runs approximately 1-3 frames per second on a standard PC platform with 2.4 GHz CPU and 3 GB memory. The parts of our tracker are generated by SLIC superpixel segmentation [19], in which the compactness is set as 50. The number of superpixels varies according to the size of the initial target, making sure the initial target includes at least 10 superpixels, and we usually set it between 200 to 600. The balance parameters for graph cut minimization α_1, α_2, λ_1 and λ_2 are 0.1, 0.4, 1.0 and 1.5 respectively, and θ_d for interaction construction is 1.5. At the step of graph matching, the parameter β is 0.4 and K is 5. For updating, the $LASVM$ is updated every 3 frames, target color histograms and graph model are updated every frame. θ_b

is 0.7, θ_a is 0.4, θ_s is 0.5 and N_f locates in the interval [3,10] according to the sequence. The local perturbation term $\mu \in [-4, 4] \times [-4, 4]$, and γ is 3 in our experiment. Moreover, we utilize the default parameters of other trackers which are provided in their papers or codes and choose the best one of 5 runs.

6.2 Experiment Analysis

Our tracker successfully adapts to large appearance and structure variations, as shown in Fig. 2. The accurate matching results with spectral technique ensures that the target center can be located robustly. Besides, the appropriate updating mechanism effectively constructs the dynamic graph on the fly, which is the prerequisite of better graph matching. We also compare our tracking results with other state-of-the-art trackers under different challenges, and we will give detailed analysis in the following.

Heavy Occlusion: The targets in the sequences *bluecar* and *lemming* undergo severe occlusion. It is difficult for TLD and HABT to locate the target because it does not have any mechanism to resist severe occlusion, but on the contrary, $\ell 1$ can precisely find the target when the blue car is severely occluded by the red car, as depicted in Fig. 4. Although MIL, IVT and Frag are robust under occlusion, they still do not have good performances in these sequence because other challenges such as shape deformation and rotation occurring before occlusion have drifted these trackers away. BHMC and SPT shrink to the non-occluded parts of the target since the inner structure information of the target has not been appropriately considered. Our dynamic graph will keep the inner structure of the target, hence our tracker robustly find the target even when some parts of the target are invisible, as demonstrated in Fig. 4.

Structure Deformation: Structure deformation is a disaster for bounding box based trackers, because the template features are totally different when severe structure deformation happens. As shown in Fig. 4, IVT, MIL, TLD, and $\ell 1$ nearly do not have satisfactory tracking results in the sequences *waterski*, *lipinski*, *yunakim* and *avatar*. Differently, Frag, HABT, BHMC and SPT have relatively better tracking results than bounding box based trackers in these sequences, because the part-based trackers are less sensitive to structure variation than holistic appearance. However, the lack of updating and scale adjustment still drives HABT and Frag to failure when quick and large structure deformation occurs. The unstable tracking of single part is the reason why BHMC cannot obtain appropriate bounding box of the target, and SPT always shrinks to local region of the target because of lacking global structure constraints. Our tracker have obvious advantage in handling structure deformation even in high frequency, since the inner structure of the target is exploited carefully and the dynamic graph robustly adapts to the structure deformation.

Illumination Variations: The frequent illumination variations in the sequence *up* lead other trackers to drift away quickly, but the incremental subspace learning method provides IVT the ability to recognize the girl even when the sunlight are blocked by the balloons for several times. Thanks to our appearance updating mechanism that *LASVM* and color histogram are incrementally

Seq.	MIL[4]	IVT[1]	TLD[14]	ℓ1[6]	Frag[9]	HABT[15]	BHMC[8]	SPT[2]	Ours
lemming	14.9	128	167	140	82.8	107	158	**7.15**	8.61
waterski	17.1	34.1	20.8	42.6	78.5	16.0	116	9.57	**8.94**
lipinski	33.8	90.8	109	46.6	50.5	30.9	14.1	12.3	**9.17**
yunakim	59.4	142	39.4	70.6	50.4	27.0	-	16.8	**15.6**
transformer	47.7	139	25.5	269	36.6	141	23.2	**10.1**	14.6
bluecar	130	83.3	48.1	90.6	92.2	80.8	186	20.1	**10.5**
avatar	107	163	162	261	139	125	18.3	18.3	**10.9**
up	150	57.3	34.0	59.0	149	66.7	55.0	37.7	**7.33**

Table 1. Comparison results of average error center location in pixel.

Seq.	Frames	MIL[4]	IVT[1]	TLD[14]	ℓ1[6]	Frag[9]	HABT[15]	BHMC[8]	SPT[2]	Ours
lemming	**1336**	1112	284	234	130	733	523	120	**1246**	**1246**
waterski	**95**	57	66	62	58	44	73	5	85	**89**
lipinski	**660**	310	35	210	135	225	105	190	90	**505**
yunakim	**571**	74	25	65	15	55	159	-	293	**501**
transformer	**124**	48	49	50	38	50	26	78	**124**	**124**
bluecar	**441**	20	74	228	263	79	45	10	149	**377**
avatar	**134**	15	7	13	16	7	15	57	51	**108**
up	**190**	38	99	60	85	53	30	39	72	**184**

Table 2. The number of successfully tracked frames.

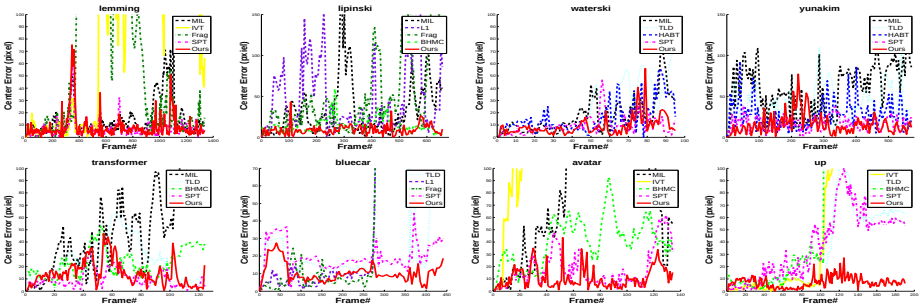


Fig. 3. Tracking results of our tracker, MIL, IVT, TLD, ℓ1, Frag, HABT, BHMC and SPT. The results of five trackers with relatively better performance are displayed.

updated with initial samples and newly coming samples, our tracker also can resist the frequent appearance variations.

Abrupt Movement and Scaling: The abrupt jumping in *waterski*, *lipinski*, *lipinski* and *avatar*, and the large scaling variation in *avatar* are challenges for many trackers. Since we collect the candidate target parts with foreground/background separation and dynamically add and delete the nodes in the target graph, our tracker will not be undermined by the abrupt movement and large scaling changes, as depicted in Fig. 3, Fig. 4, Table 1 and Table 2.

7 Conclusion

A novel online part-based tracker is proposed in this paper, in which the parts semantically generated by oversegmentation and the interaction between parts

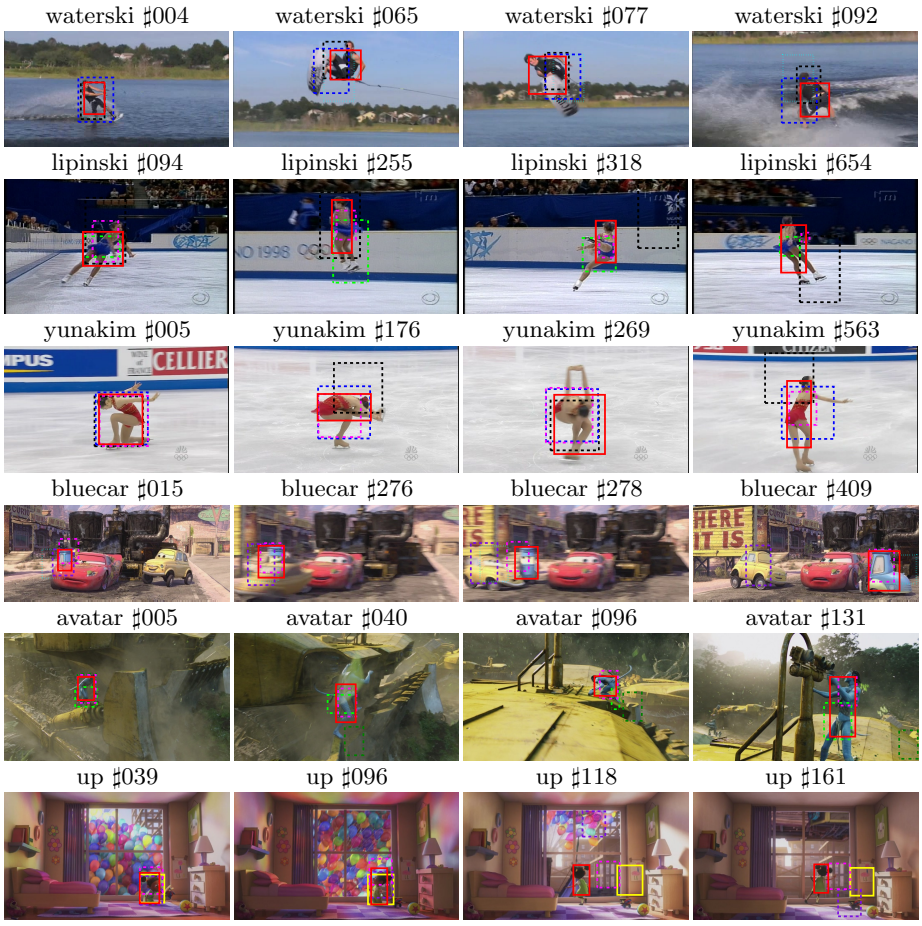


Fig. 4. The results of our tracker, MIL, IVT, TLD, $\ell 1$, Frag, HABT, BHMC, and SPT are depicted as red, black, yellow, cyan, purple, dark green, blue, light green, and magenta rectangles respectively.

are modeled as a dynamic graph. The target tracking is interpreted as matching the candidate graph to the target graph. The spatial prior with MRF helps us to separate the candidate target parts from the background and form the candidate graph. The matching is also modeled as an undirect graph, whose optimization is solved with spectral technique. This holistic target tracking mechanism owns more robustness because of the introduction of structure information.

Acknowledgement This work was supported by the Chinese National Natural Science Foundation Project #61070146, #61105023, #61103156, #61105037, #61203267, National IoT R&D Project #2150510, Chinese Academy of Sciences Project No. KGZD-EW-102-2, European Union FP7 Project #257289 (TABULA RASA http://www.tabularasa-euproject.org), and AuthenMetric R&D Funds.

References

1. Lim, J., Ross, D.A., Lin, R.S., Yang, M.H.: Incremental learning for visual tracking. In: NIPS. (2004)
2. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: ICCV. (2011) 1323–1330
3. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR (1). (2006) 260–267
4. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 1619–1632
5. Tian, M., Zhang, W., Liu, F.: On-line ensemble svm for robust object tracking. In: ACCV (1). (2007) 355–364
6. Mei, X., Ling, H.: Robust visual tracking using $\ell 1$ minimization. In: ICCV. (2009) 1436–1443
7. Liu, B., Huang, J., Yang, L., Kulikowski, C.A.: Robust tracking using local sparse appearance model and k-selection. In: CVPR. (2011) 1313–1320
8. Kwon, J., Lee, K.M.: Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In: CVPR. (2009) 1208–1215
9. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR (1). (2006) 798–805
10. Ren, X., Malik, J.: Tracking as repeated figure/ground segmentation. In: CVPR. (2007)
11. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010) 1627–1645
12. Quattoni, A., Collins, M., Darrell, T.: Conditional random fields for object recognition. In: NIPS. (2004)
13. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26** (2004) 1124–1137
14. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N learning: Bootstrapping binary classifiers by structural constraints. In: CVPR. (2010) 49–56
15. Shahed, S.M.N., Ho, J., Yang, M.H.: Online visual tracking with histograms and articulating blocks. Computer Vision and Image Understanding **114** (2010) 901–914
16. Yang, M., Wu, Y., Lao, S.: Intelligent collaborative tracking by mining auxiliary objects. In: CVPR (1). (2006) 697–704
17. Tsai, D., Flagg, M., Rehg, J.M.: Motion coherent tracking with multi-label mrf optimization. In: BMVC. (2010) 1–11
18. Grundmann, M., Kwatra, V., Han, M., Essa, I.A.: Efficient hierarchical graph-based video segmentation. In: CVPR. (2010) 2141–2148
19. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Ssstrunk, S.: SLIC Superpixels. Technical report, EPFL (2010)
20. Bordes, A., Ertekin, S., Weston, J., Bottou, L.: Fast kernel classifiers with online and active learning. Journal of Machine Learning Research **6** (2005) 1579–1619
21. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV. (2005) 1482–1489
22. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: CVPR. (2010) 723–730