Robust Deformable and Occluded Object Tracking With Dynamic Graph

Zhaowei Cai, Longyin Wen, Zhen Lei, *Member, IEEE*, Nuno Vasconcelos, *Senior Member, IEEE*, and Stan Z. Li, *Fellow, IEEE*

***Abstract*— While some efforts have been paid to handle deformation and occlusion in visual tracking, they are still great challenges. In this paper, a dynamic graph-based tracker (DGT) is proposed to address these two challenges in a unified framework. In the dynamic target graph, nodes are the target local parts encoding appearance information, and edges are the interactions between nodes encoding inner geometric structure information. This graph representation provides much more information for tracking in the presence of deformation and occlusion. The target tracking is then formulated as tracking this dynamic undirected graph, which is also a matching problem between the target graph and the candidate graph. The local parts within the candidate graph are separated from the background with Markov random field, and spectral clustering is used to solve the graph matching. The final target state is determined through a weighted voting procedure according to the reliability of part correspondence, and refined with recourse to a foreground/background segmentation. An effective online updating mechanism is proposed to update the model, allowing DGT to robustly adapt to variations of target structure. Experimental results show improved performance over several state-of-the-art trackers, in various challenging scenarios.**

***Index Terms*— Visual tracking, dynamic graph, graph matching, deformation, occlusion.**

I. Introduction

VISUAL tracking is an important problem in computer vision, with applications in video surveillance, human-computer interaction, behavior analysis, etc. The development of a robust visual tracking algorithm is difficult due to challenges such as shape deformation, occlusion, and appearance variability. In particular, challenging deformation and occlusion are ubiquitous in tracking problems. While many trackers [2], [15], [29] have considered occlusion, only a few works [20], [40] have addressed the problem of shape deformation. In this paper, we approach the two challenges in a unified framework, dynamic graph based tracking, because graph representation intuitively owns the power to recognize the geometric deformable target, and to accurately localize the occluded target with the other unoccluded parts of the target.

Recent trackers achieve high tracking accuracy and robustness mainly through three aspects: feature, appearance model and structure information. Features commonly used with different properties include pixel values [23], color [2], [8], [30], [40], and texture descriptors [3], [14]. The appearance model is used to characterize the target, such as color distribution [8], [30], subspaces [23], [42], Support Vector Machine [37], Boosting [3], [14], [19] and sparse representation [24], [29], [45]. Finally, an increasing number of trackers capture the structure information [2], [16], [20], [36], [41], [43], similar to the popular approaches in object detection [12], recognition [31], etc. The inner structure can be particularly useful for tracking when deformation and occlusion are prevalent.

In this paper, we design a dynamic graph based tracker (DGT), which exploits the inner geometric structure information of the target. This structure information is generated by oversegmenting the target into several parts (superpixels) and then modeling the interactions between neighboring parts. Both the appearances of local parts and their relations are incorporated into a dynamic undirected graph. The tracking problem is then posed as the tracking of this undirected graph, which is also a matching problem between the target graph $\mathcal{G}(V, E)$ and the candidate graph $\mathcal{G}'(V', E')$. The candidate target parts are obtained, at the beginning of every tracking step, by foreground/background separation, using Markov Random Field. The undirected candidate graph is then assembled with these obtained candidate parts and their geometric interactions. At the step of graph matching, motion, appearance and geometric constraints are exploited to produce a less-noisy and more discriminative graph affinity matrix. The optimal matching from candidate graph to target graph is interpreted as finding the main cluster from the affinity matrix, using spectral technique [22]. The location of the target is determined by a voting process, where successfully matched parts vote for a particular location with strength proportional to the reliability of their correspondence. Finally, the target location and scale are adjusted with recourse to a foreground/background segmentation.

Manuscript received September 30, 2013; revised July 14, 2014 and September 29, 2014; accepted October 14, 2014. Date of publication October 23, 2014; date of current version November 17, 2014. This work was supported in part by the U.S. National Science Foundation under Grant CCF-0830535 and Grant IIS-1208522, in part by the Chinese Academy of Sciences, Beijing, China, under Project KGZDEW-102-2, in part by the National Natural Science Foundation of China under Project 61203267, Project 61375037, and Project 61473291, and in part by the National Science and Technology Support Program under Project 2013BAK02B01. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Erhardt Barth.

Z. Cai and N. Vasconcelos are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: zwcai@ucsd.edu; nuno@ucsd.edu).

L. Wen, Z. Lei, and S. Z. Li are with the National Laboratory of Pattern Recognition, Center for Biometrics and Security Research, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lywen@nlpr.ia.ac.cn; zlei@nlpr.ia.ac.cn; szli@nlpr.ia.ac.cn).

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.2014.2364919

The main contributions of this work are as follows[1]:

- We represent the target as a graph and formulate tracking as graph matching problem, between a candidate graph and the target graph. This graph representation has advantage of jointly accounting for deformation and occlusion.
- The geometric structure information is exploited throughout the proposed target representation, selection of candidate parts, graph matching and target location processes. The geometric structure provides additional useful information besides appearance for visual tracking.
- We construct the affinity matrix for matching based on motion, appearance and geometric constraints, which efficiently suppress the noise and decreases the complexity.

The remainder of the paper is organized as follows. Section II reviews related works. Section III discusses the target representation by an undirected graph and the formulation of tracking as graph matching. In details, Section IV describes the procedure of constructing the candidate graph, local parts correspondence with spectral matching is discussed in Section V, Section VI describes the target location strategy and Section VII the online updating. Experimental results and discussions are presented in Section VIII, and some conclusions in Section IX.

II. Related Works

Most tracking approaches represent the target as a bounding box template. An incremental subspace is modeled in [23] to represent the target, which shows robustness against illumination variations. Some trackers [14], [37] employ binary classifier (e.g. SVM or Boosting) to model the difference between target and background. To enhance robustness, [21] decomposes a tracker into several small trackers, [42] proposes a combination of temporal and spatial context information, [28] relies on saliency, and [18] introduces a detector in the tracking process. None of these methods consider either deformation or occlusion.

Some other methods have made an effort to address occlusion. Adam et al. [2] have shown that evenly segmenting the target into horizontal and vertical patches can improve robustness to partial occlusion. [24], [29], [45] used a sparse representation to reconstruct the target from a set of appearance features. This is relatively insensitive to occlusion since they use a large set of trivial templates. [3] adopted a multiple instance learning strategy to minimise the effects of occlusion during the learning of a target/background classifier. Grabner et al. [15] introduced context information to overcome wholly occlusion problem, and the results seem pretty good. Nevertheless, these methods ignore the deformation problem.

Part-based model is a sensible solution to deformation challenge. Several strategies have been proposed to generate local parts in visual tracking: equal division [2], [7], [42], manual partition [34], oversegmentation [16], [32], [40], [41], kernel displacement [44], key-point [20], [36], etc. These methods have different merits and limitations. For example, although the target is represented as manually located parts in [34], only limited structure information is captured and the relation between local parts does not evolve over time. The strategy in [2] lacks adaptivity needed to handle large structure deformation. [20], [36] generate local parts with the help of key-point mechanism. This improves robustness to scale and rotation. On the other hand, the parts (superpixels) generated by oversegmentation in [16], [32], [40], and [41] enables a richer characterization of the target, allowing finer part discrimination. Finally, the parts (attentional regions) selected by kernel displacement in [44] are psychologically meaningful, sensitive to motion and reliable for tracking.

The representation of the target from the local parts is a critical issue for part-based tracking. In [40], the target is represented as a collection of individual parts. It is assumed that there is no dependence between parts, nor any dependence between parts and target. This is computationally efficient but results in the lack of any structural constraints. Another computationally feasible representation is the star model, which encodes dependence between parts and the target center, but no inter-part dependence. Although this model has been shown effective for tracking [2], [20], [34], [41], [43], [44], these trackers only utilize limited geometric structure information. Besides, the connections between target center and parts are usually rigid, reducing the flexibility against deformation. To address this problem, the star model in [12] assigns costs on the geometric connections, allowing real deformation.

To sufficiently exploit the inner geometric structure information for visual tracking, undirected graph is an appropriate choice [16], [36]. Tang et al. [36] introduced attributed relational feature graph (ARG) into tracking, whose nodes represent appearance descriptors of parts while edges account for part relationships. The optimal state is found in a probabilistic framework, and graph matching optimized by Markov Random Field is used to compute the likelihood of sample targets. An alternative ARG tracker has been proposed by Graciano et al. [16]. The recognition is performed by inexact graph matching, which consists of finding an approximate homomorphism between ARGs derived from an input video and a model image. Like these approaches, the proposed dynamic graph tracker represents the target as an undirected graph, and formulate tracking as graph matching. However, we exploit graph representation to jointly account for deformation and occlusion challenges. Moveover, our tracker performs very well in long-term sequences and adapts to large target structure variations because of the effective online learning strategies.

Video segmentation [17] also relates to our work, since the separation of candidate target parts from background in our tracker is accomplished by segmentation in terms of superpixel. Segmentation is widely used in tasks such as detection [13], recognition [38] and tracking [26], [32], [40], because segmentation can 1) introduce prior for searching, 2) obtain semantic region and 3) extract edge information. Since, in our work, the goal of segmentation is simply to produce candidate parts, not solve the tracking problem itself, we choose to use simple segmentation methods.

[1] Part of this work was presented in [6].

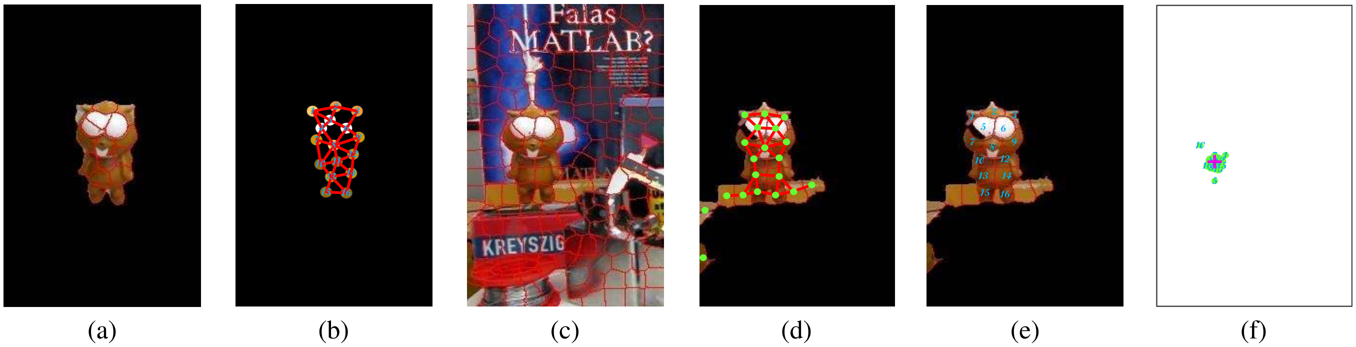


Fig. 1. The framework illustration of our tracker. (a) is the target initialization, (b) the target graph, (c) the superpixel oversegmentation in the tracking window, (d) the segmented candidate local parts and the candidate graph, (e) the correspondence results, and (f) the illustration of our weighted voting, where green circle is the voted center for single correspondence and the magenta cross is the final voted center. The bigger the green circle is, the more reliable the correspondence is. The numbers in (e) and (f) correspond to the node indices in (b).

Rough segmentation at the level of superpixel is enough because the correspondence of local parts is more important.

III. Tracking With Graph Representation

The goal of visual tracking is to sequentially identify the location of the target in a set of video frames. As illustrated in Fig. 1(a), the target is represented as a set of local parts $\{T_i\}_{i=1}^{n_P}$, obtained by [1]. Part T_i is characterized by a feature f_i, a location l_i, and its offset $R_i = l_c - l_i$ to the target center l_c. A popular way to track the individual local parts is within Maximum a Posterior (MAP) estimation. Instead, we view the tracking of individual local parts as a matching problem here, after obtaining the set of candidate local parts $\{T_{i'}\}_{i'=1}^{n_Q}$ from current frame with some operators, such as oversegmentation [1] and SIFT [25]. The local part T_i is successfully tracked as $T_{i'}$ when the correspondence $c_{ii'} = (T_i, T_{i'})$ between them is true. It is common to represent the correspondence by an assignment matrix $\mathbf{Z} \in \{0,1\}^{n_P \times n_Q}$, where $z_{ii'} = 1$ means T_i corresponds to $T_{i'}$, otherwise, $z_{ii'} = 0$. The assignment matrix $\mathbf{Z}$ is reshaped into a row assignment vector $\mathbf{z} \in \{0,1\}^{n_P n_Q}$. Then, the optimal correspondence result will be obtained by finding $\mathbf{z}^*$ that maximizes a score function $S(\mathbf{z})$:

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} S(\mathbf{z}) \tag{1}$$

$$s.t. \quad \mathbf{Z}\mathbf{1}_{n_Q \times 1} \leq \mathbf{1}_{n_P \times 1}, \quad \mathbf{Z}^T\mathbf{1}_{n_P \times 1} \leq \mathbf{1}_{n_Q \times 1}$$

where $\mathbf{1}_{n \times 1}$ denotes all-ones column vector with size n. The optimization constrains guarantee a one-to-one matching between $\{T_i\}_{i=1}^{n_P}$ and $\{T_{i'}\}_{i'=1}^{n_Q}$.

For unary appearance information only, we construct the unary affinity matrix $\mathbf{A}_0 \in \mathbb{R}^{n_P \times n_Q}$ between $\{T_i\}_{i=1}^{n_P}$ and $\{T_{i'}\}_{i'=1}^{n_Q}$. Each entry in $\mathbf{A}_0$ indicates the appearance similarity between two local parts:

$$a_{ii'} = \Omega_1(c_{i,i'}) = \Omega_1(T_i, T_{i'}) \tag{2}$$

Then, the score function is

$$S(\mathbf{z}) = \mathbf{z}^T\mathbf{a} \tag{3}$$

where $\mathbf{a} \in \mathbb{R}^{n_P n_Q}$ is a column-wise vectorized replica of $\mathbf{A}_0$. Although the tracking result can be obtained through (1) with unary appearance information only, the result is not robust because unary appearance information is not reliable enough, especially for color histogram based local parts. Therefore, we integrate the pairwise mutual relation between local parts, and then construct the affinity matrix $\mathbf{A} \in \mathbb{R}^{n_P n_Q \times n_P n_Q}$:

$$\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2 \tag{4}$$

where $\mathbf{A}_1 \in \mathbb{R}^{n_P n_Q \times n_P n_Q}$ is a diagonal matrix whose diagonal vector is $\mathbf{a}$ in (3), and $\mathbf{A}_2 \in \mathbb{R}^{n_P n_Q \times n_P n_Q}$ is the pairwise affinity matrix. Each entry in $\mathbf{A}_2$ indicates the relation between two correspondences:

$$\begin{aligned} a_{\hat{i},\hat{j}} &= a_{(i-1)n_Q+i',(j-1)n_Q+j'} \\ &= \Omega_2(c_{ii'}, c_{jj'}) = \Omega_2(T_i, T_{i'}, T_j, T_{j'}) \end{aligned} \tag{5}$$

Then, the score function in (3) is reformulated as

$$S(\mathbf{z}) = \mathbf{z}^T\mathbf{A}\mathbf{z} \tag{6}$$

The optimization of the above score function is a conventional graph matching problem between the target graph $\mathcal{G}(V, E)$ whose vertices set is $\{T_i\}_{i=1}^{n_P}$, Fig. 1(b), and the candidate graph $\mathcal{G}'(V', E')$ whose vertices set is $\{T_{i'}\}_{i'=1}^{n_Q}$, Fig. 1(d). After the optimal correspondences being determined, the optimal tracking results of the target parts are achieved.

Based on the above discussion, we view part-based object tracking as graph matching problem. In this framework, the appearance and inner structure of the target can be well integrated. Instead of representing the target as the collection of local parts or star model, we represent the target as an undirected graph, as shown in Fig. 1(b). Given the target graph $\mathcal{G}(V, E)$ and the candidate graph $\mathcal{G}'(V', E')$, our goal is to find the optimal correspondence between them, and to determine the optimal target state based on the correspondence results, as shown in Fig. 1(e) and (f).

IV. Candidate Graph Construction

Given the current frame, a candidate graph is needed to be constructed to match the target graph. The simplest way is to connect all of the parts in the tracking window Fig. 1(c) without any pre-processing. However, a lot of noise will be introduced and the complexity will be exponentially high in that way. Therefore, we rely on a pre-processing strategy to construct a less-noisy candidate graph.

A. Candidate Local Parts Collection

In this work, we use superpixels as parts. A set of parts $\{T_p\}$ is first extracted from the current frame using *Simple Linear Iterative Clustering* (SLIC) [1], as shown in Fig.1(c). A rough foreground/background separation is then used as the pre-processing strategy to collect candidate target parts from $\{T_p\}$. This is based on a Markov Random Field (MRF) energy function:

$$E(\mathbf{B}) = \sum_{p \in S} D_p(b_p) + \sum_{p,q \in N} V_{p,q}(b_p, b_q) \tag{7}$$

where $\mathbf{B} = \{b_p | b_p \in \{0,1\}, p \in S\}$ is the labeling of superpixels set $\{T_p\}$, b_p an indicator function for part T_p ($b_p = 1$ if T_p belongs to foreground and $b_p = 0$ otherwise), $D_p(b_p)$ a unary potential associated with superpixel T_p, and $V_{p,q}(b_p, b_q)$ a pairwise potential for interacting superpixels T_p and T_q. S is the set of superpixels in the tracking window, and N the set of pairs of interacting superpixels with shared edges (red lines in Fig. 1(c)). The minimization of (7) can be found with several algorithms [35], and Graph Cut [5] is used here because its high running efficiency fits in tracking problem.

In (7), the unary potential $D_p(b_p)$ is a weighted combination

$$D_p(b_p) = \alpha D_p^g(b_p) + D_p^d(b_p) \tag{8}$$

of a generative color histogram potential $D_p^g(b_p)$ and a discriminant SVM classifier potential $D_p^d(b_p)$. $\alpha = 0.1$ is a constant to balance the influences of the two potential terms. The generative potential is of the form

$$D_p^g(b_p) = \begin{cases} -\frac{1}{N_p}\sum_{i=1}^{N_p} \log P(C_i|\mathcal{H}^b) & b_p = 1 \\ -\frac{1}{N_p}\sum_{i=1}^{N_p} \log P(C_i|\mathcal{H}^f) & b_p = 0 \end{cases} \tag{9}$$

where $\mathcal{H}^f$ and $\mathcal{H}^b$ are normalized RGB color histograms of the target and the background, respectively, C_i is the RGB value of pixel i, and N_p the number of pixels in superpixel T_p. $P(C_i|\mathcal{H})$ is the probability of C_i within histogram $\mathcal{H}$. The discriminant potential is the classification score of an online SVM [4] classifier trained from RGB color features extracted from the target and the background superpixels,

$$D_p^d(b_p) = \begin{cases} \lambda\widehat{y}(f_p) & \widehat{y}(f_p) \geq 0, \quad b_p = 1 \\ 1 - \lambda\widehat{y}(f_p) & \widehat{y}(f_p) \geq 0, \quad b_p = 0 \\ \widehat{y}(f_p) & \widehat{y}(f_p) < 0, \quad b_p = 1 \\ 1 - \widehat{y}(f_p) & \widehat{y}(f_p) < 0, \quad b_p = 0 \end{cases} \tag{10}$$

where $\widehat{y}(f_p) = \mathbf{w} \cdot \Phi(f_p) + b$ is the SVM discriminant, and f_p is the color feature of T_p. $\lambda = 15$ is a constant that strengthens the influence of SVM classifier when it classifies T_p as foreground, because we want to keep true foreground superpixels as many as possible. $V_{p,q}(b_p, b_q)$ captures the discontinuity between two neighboring superpixels which is viewed as smoothness term:

$$V_{p,q}(b_p, b_q) = \exp\left\{-D(f_p, f_q)\right\} \tag{11}$$

where $D(,)$ is the $\mathcal{X}^2$ distance between color features throughout this paper. It encourages the target to be a collection of connected parts of similar appearance, as shown in Fig. 1(d). Finally, the candidate part set is collected, $\{T_{i'}\}_{i'=1}^{n_Q} = \{T_p | b_p = 1\}$.

B. Graph Construction

There are mainly three popular methods for graph construction: the ε-neighborhood graph, the k-nearest neighbor graph, and the fully connected graph [39]. Given the candidate part set $\{T_{i'}\}_{i'=1}^{n_Q}$, we adopt the method of ε-neighborhood to construct the candidate graph $\mathcal{G}'(V', E')$, whose vertices set V' and edges set E' are

$$\begin{aligned} V' &= \{T_{i'}\}_{i'=1}^{n_Q} \\ E' &= \{e_{i'j'} | dist(T_{i'}, T_{j'}) \leq \varepsilon\} \end{aligned} \tag{12}$$

where $dist(,)$ is the geometric distance between two local parts, and $\varepsilon = r\theta_d$. $\theta_d = 2$ is the constant, and $r = \sqrt{W \cdot H / N_s}$ is the approximate diameter of superpixels throughout this paper, 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. The resulting candidate graph is illustrated in Fig. 1(d). It could be noted that ε is fixed, in which case the graph inconsistency may arise when the target undergoes large scale variation. Since the target graph keeps updating, the problem could be effectively overcome.

V. Local Parts Correspondence With Spectral Matching

The core component of the proposed part-based tracker is the determination of part correspondences across frames. A good correspondence result prevents our tracker from being dominated by segmentation. Even when the output of the foreground/backbround separation step contains some noise, the local parts correspondence step still can determine the most similar subgraph to the target graph, and the noisy parts have small influence on the final target location.

A. Affinity Matrix Construction

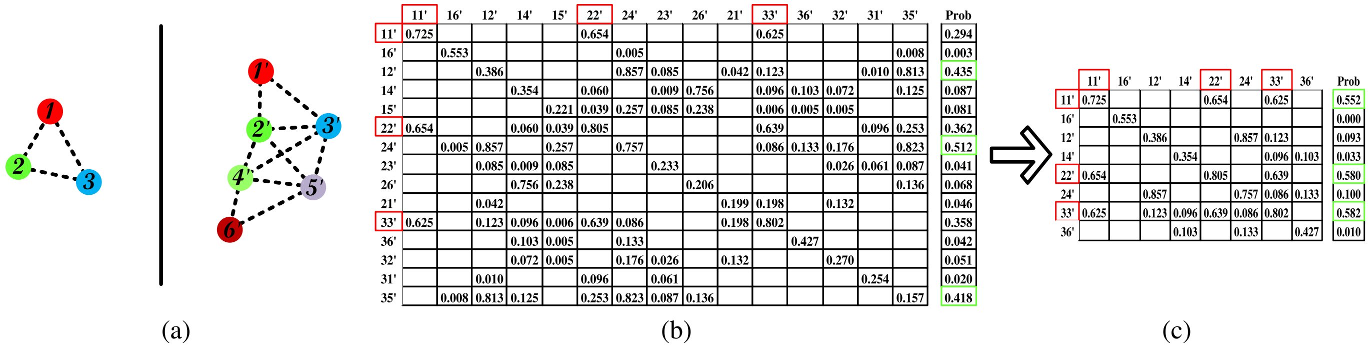
As discussed in Section III, the affinity matrix $\mathbf{A}$ consists of two parts: the unary appearance affinity matrix $\mathbf{A}_1$ and the pairwise relation affinity matrix $\mathbf{A}_2$. The unary affinity in (2) encodes vertex-to-vertex appearance similarity:

$$\Omega_1(c_{ii'}) = \Omega_1(T_i, T_{i'}) = \exp\left\{-\frac{1}{\varepsilon_1^2}(D(f_i, f_{i'}))^2\right\} \tag{13}$$

where $\varepsilon_1 = 1/\sqrt{2}$ is the Gaussian kernel bandwidth. The pairwise affinity in (5) encodes edge-to-edge geometric relation:

$$\begin{aligned} \Omega_2(c_{ii'}, c_{jj'}) &= \Omega_2(T_i, T_{i'}, T_j, T_{j'}) \\ &= \exp\left\{-\frac{1}{\varepsilon_2^2}||(l_i - l_j) - (l_{i'} - l_{j'})||_2^2\right\} \end{aligned} \tag{14}$$

where $\varepsilon_2 = r/\sqrt{2}$ is the Gaussian kernel bandwidth and l_i the location of part T_i. Ω_1 indicates how well an individual assignment $c_{ii'}$ is matched, and Ω_2 denotes how well the two geometrically pairwise assignments $c_{ii'}$ and $c_{jj'}$ are compatible. Color appearance information encoded in Ω_1 is usually weak, especially for superpixels. However, the geometric information encoded in Ω_2 provides much more complementary information besides appearance for graph matching, as shown in Fig. 3. Also note that although Ω_2 does not encourage large scaling and rotation, it is elastic on the geometric relation,



(b)

	11'	16'	12'	14'	15'	22'	24'	23'	26'	21'	33'	36'	32'	31'	35'	Prob
11'	0.725					0.654					0.625					0.294
16'		0.553					0.005								0.008	0.003
12'			0.386				0.857	0.085		0.042	0.123			0.010	0.813	0.435
14'				0.354		0.060		0.009	0.756		0.096	0.103	0.072		0.125	0.087
15'					0.221	0.039	0.257	0.085	0.238		0.006	0.005	0.005			0.081
22'	0.654			0.060	0.039	0.805					0.639			0.096	0.253	0.362
24'		0.005	0.857		0.257		0.757				0.086	0.133	0.176		0.823	0.512
23'			0.085	0.009	0.085			0.233					0.026	0.061	0.087	0.041
26'				0.756	0.238				0.206						0.136	0.068
21'			0.042							0.199	0.198		0.132			0.046
33'	0.625		0.123	0.096	0.006	0.639	0.086			0.198	0.802					0.358
36'				0.103	0.005		0.133					0.427				0.042
32'				0.072	0.005		0.176	0.026		0.132			0.270			0.051
31'			0.010			0.096		0.061						0.254		0.020
35'		0.008	0.813	0.125		0.253	0.823	0.087	0.136						0.157	0.418

(c)

	11'	16'	12'	14'	22'	24'	33'	36'	Prob
11'	0.725				0.654		0.625		0.552
16'		0.553							0.000
12'			0.386			0.857	0.123		0.093
14'				0.354			0.096	0.103	0.033
22'	0.654				0.805		0.639		0.580
24'			0.857			0.757	0.086	0.133	0.100
33'	0.625		0.123	0.096	0.639	0.086	0.802		0.582
36'				0.103		0.133		0.427	0.010

Fig. 2. The construction of the affinity matrix with noisy input. (a) is the two graphs that are needed to be matched, 123-1′2′3′ is the correct matching, and the color represents appearance. (b) is the affinity matrix built by 5-nearest neighbor way and (c) built by combing 5-nearest neighbor way and 0.3-neighborhood way. The red borders represent right correspondences, and the green borders represent the selected correspondences by spectral matching. The rightmost column of (b) and (c) is the spectral matching probability. The affinity matrix in (c) is of lower complexity and more resistance to noise than the one in (b). The spectral matching result in (b) is wrong, but it is correct in (c).

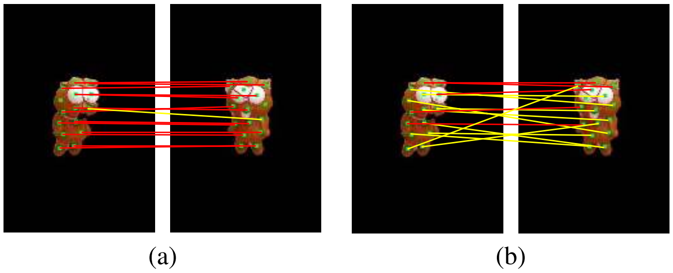


Fig. 3. (a) is matching results of $\mathbf{A}$ optimized by spectral technique, and (b) the matching results of $\mathbf{A}_1$ optimized by greedy strategy, where red lines mean good correspondences and yellow lines represent bad ons.

enabling the model to tolerate scaling and rotation to some degree.

The dimension of $\mathbf{A}$ is $n_P n_Q \times n_P n_Q$, which means any vertex in $\mathcal{G}(V, E)$ can be potentially matched to any vertex in $\mathcal{G}'(V', E')$, and vice versa. However, the overwhelming number of potential assignments is unreasonable in matching task [10]. To reduce complexity, we need to prune $\mathbf{A}$.

Similar to the graph construction methods in Section IV-B, there are also three main ways to select potential assignments for single vertex: the ε-neighborhood way, the k-nearest neighbor way, and the fully connected way. But at this time the neighborhood measure is the feature distance instead of the geometric distance. The first two choices will shrink the size of $\mathbf{A}$, but the last one not. The ε-neighborhood way will select many potential assignments for parts whose appearance is not discriminative, and only a few ones for discriminative parts. This may make the probability of the undiscriminative correspondence larger over that of the discriminative correspondence in matching process. The main alternative in the literature, k-nearest neighbor, is also found to have problems: the requirement that every vertex has k potential assignments leads to the selection of many potential assignments that are clearly impossible. To overcome the problems of these two strategies, we combine them. Besides, we introduce movement constraints. Only if $T_{i'}$ and T_i obey the motion constraint, will they potentially be matched, which is $||l_i - l_{i'}||_2 \le r\delta$, where $\delta = 3$ is a constant. With the introduction of motion constraint, graph matching fits into tracking better.

Based on the above discussion, the potential assignments set $\mathcal{F}_i$ for every single local part is

$$\mathcal{F}_i = \left\{ c_{ii'} | i' \in \mathcal{N}_k^i,\ \Omega_1(T_i, T_{i'}) \ge \eta,\ ||l_i - l_{i'}||_2 \le r\delta \right\} \quad (15)$$

where $\mathcal{N}_k^i$ is the k-nearest neighborhood of part T_i in feature space, $k = 5$ and $\eta = 0.3$. As displayed in Fig. 2(c), with this strategy, a discriminative and less-noisy affinity matrix with dimension $d = \sum_{i=1}^{n_P} |\mathcal{F}_i|$ is constructed, and our tracker will be protected from drifting across structurally similar distracters, as happens in Fig. 2(b).

After obtaining the whole potential assignment set $\{\mathcal{F}_i\}_{i=1}^{n_P}$, we need to fill in the non-diagonal entries in the affinity matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$. Usually, $\mathbf{A}$ is sparse because only the assignments that are compatible with each other will support each other during the matching process. At first, the pairwise affinity $\Omega_2(c_{ii'}, c_{jj'})$ is computed with (14) if there exists edges e_{ij} and $e_{i'j'}$. However, this constraint is too weak such that many incompatible assignments will support each other too. For example, 16′ and 24′ in Fig. 2(a) are two obvious incorrect assignments, but they still support each other in the affinity matrix in Fig. 2(b). While a single inappropriate support will not corrupt the affinity matrix, their existence does matter if there exist too many of them. Therefore, another two constraints (from distance and intersection angle) are adopted here to filter out those noisy entries:

$$\begin{aligned} ||(l_i - l_j) - (l_{i'} - l_{j'})||_2 &< r\theta_{dist} \\ \arccos \frac{(l_i - l_j) \cdot (l_{i'} - l_{j'})}{||(l_i - l_j)||_2 ||(l_{i'} - l_{j'})||_2} &< \pi\theta_{angle} \end{aligned} \quad (16)$$

where $\theta_{dist} = 2$ and $\theta_{angle} = \frac{5}{8}$ in our experiments. It is reasonable to constrain the distance and rotation angle changes here, since the task is tracking and the target graph keeps updating sequentially. Most of the reasonable supporting correspondence pairs will meet these constraints. With their help, some inappropriate supports will be filtered out, the one between 16′ and 24′ for instance. Fig. 2(c) illustrates how a less-noisy affinity matrix is helpful for robust graph matching.

B. Spectral Matching

The optimization problem of (1) with the score function (6) can be solved by various approaches, such as quadratic programming [27], spectral technique [9], [22], etc. In this work, we adopt the spectral method, mostly for its reduced complexity. It equates the graph matching solution to finding the main cluster in the assignment graph. We start by considering the following program:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \frac{\mathbf{x}^T \mathbf{A}\mathbf{x}}{\mathbf{x}^T \mathbf{x}} \quad s.t. \quad \mathbf{x}^T \mathbf{x} = 1 \tag{17}$$

This optimization can be conventionally solved by eigenvector technique by Raleigh's ratio theorem, and $\mathbf{x}^*$ is well known to be the leading eigenvector of $\mathbf{A}$. Since the unary and pairwise affinity components of (13) and (14) are based on Gaussian kernels, and the graph is undirected, guaranteeing that $\mathbf{A}$ is symmetric and nonnegative, $\mathbf{x}^*$ is known to exist and be nonnegative, by Peron-Frobenius theorem [22].

$\mathbf{x}^* \in \mathbb{R}^d$ is a soft assignment vector, instead of the binary assignment indicator vector $\mathbf{z}^* \in \mathbb{R}^d$. Based on the discussions in [10] and [22], every element in $\mathbf{x}^*$ can be viewed as the probability $P(c_{ii'})$ of the corresponding assignment $c_{ii'}$. But it does not lead the optimized results to be sparse and binary due to the l_2-norm constraint $||\mathbf{x}||_2 = 1$. Therefore, we need a discretization strategy to map $\mathbf{x}^*$ into $\mathbf{z}^*$. A simple greedy approach is enough here, whose performance has been demonstrated in [22]. Fig. 3 shows the matching performance is dramatically improved by including pairwise geometric relation besides unary appearance information.

VI. Reliable Target Location

Given the part correspondence set $\mathcal{C} = \{ii'|z_{ii'} = 1\}$, it remains to determine the location l_c and scale $s = (w, h)$ of the target, where w is the target width and h its height. Every single correspondence $c_{ii'}$ can be seen as a single tracker, and n successful correspondences mean n trackers. The target center for $c_{ii'}$ is

$$l_c^{ii'} = l_{i'} + R_{i'} \tag{18}$$

where the offset $R_{i'} = R_i$ because T_i and $T_{i'}$ are corresponding parts. It must take into account that not all correspondence are equally reliable, thus the goal of this section is to determine the robust target state based on the correspondences, boosting the reliable ones and suppressing the other noisy ones. The reliability of the correspondence between T_i and $T_{i'}$ is measured with

$$\omega'_{ii'} = \Omega_1(c_{ii'})\Big(\frac{1}{|\mathcal{N}_i|}\sum_{jj'\in\mathcal{C}|j\in\mathcal{N}_i} \Omega_2(c_{ii'}, c_{jj'})\pi_{jj'} + \frac{1}{|\mathcal{N}_{i'}|}\sum_{jj'\in\mathcal{C}|j'\in\mathcal{N}_{i'}} \Omega_2(c_{ii'}, c_{jj'})\pi_{jj'}\Big) \tag{19}$$

where $j \in \mathcal{N}_i$ means j is in the neighborhood of i and the edge e_{ij} exists. Ω_1 and Ω_2 represent unary and pairwise information in (13) and (14), then the reliability is from the views of appearance and structure. $\pi_{jj'}$ indicates how reliable the correspondence $c_{jj'}$ is:

$$\pi_{jj'} = \Omega_1(c_{jj'})\Big(\frac{1}{|\mathcal{N}_j| - 1}\sum_{kk'\in\mathcal{C}|k\in\mathcal{N}_j\backslash i} \Omega_2(c_{jj'}, c_{kk'}) + \frac{1}{|\mathcal{N}_{j'}| - 1}\sum_{kk'\in\mathcal{C}|k'\in\mathcal{N}_{j'}\backslash i'} \Omega_2(c_{jj'}, c_{kk'})\Big) \tag{20}$$

where $\mathcal{N}_j\backslash i$ is the set of neighbors of j other than i. $\pi_{jj'}$ makes the estimation of $\omega'_{ii'}$ more robust. The location of the target is then estimated through a weighted vote from all correspondences, where the contribution of each correspondence is proportional to its reliability,

$$l_c = \sum_{c_{ii'}} \omega_{ii'} l_c^{ii'} \tag{21}$$

where $\omega_{ii'} = \omega'_{ii'}/\sum_{c_{ii'}} \omega'_{ii'}$ is the normalized weight of $c_{ii'}$. Since bad correspondences are usually incompatible with their neighboring correspondences, they have low reliability under (19) and thus small weights. As shown in Fig. 1(f) and Fig. 5, this allows the reliable localization of the target in the presence of correspondence noise.

This estimate of target location is then fine-tuned with the foreground/background segmentation of the current frame, discussed in Section IV. We fine-tune the target center with a small perturbation μ and the target scale s, so that the bounding box will cover as much foreground as possible. The optimal scale s^* and μ^* are

$$(\mu^*, s^*) = \arg\max_{\mu, s}\Big\{\gamma \cdot N^{mat}(l_c + \mu, s) + N^{pos}(l_c + \mu, s) - N^{neg}(l_c + \mu, s)\Big\} \tag{22}$$

where $N^{mat}(l_c + \mu, s)$, $N^{pos}(l_c + \mu, s)$ and $N^{neg}(l_c + \mu, s)$ are the numbers of 1) positive pixels within the matched superpixels, 2) positive pixels outside the matched superpixels, and 3) negative pixels, respectively, in the bounding box of scale s and location $l_c + \mu$. $\gamma = 3$ is a constant that emphasizes the influence of matched parts. Then the final target center and scale are, respectively, $l_c^* = l_c + \mu^*$ and s^*.

VII. Online Update

Model update is an important component of any visual tracking system, since the target frequently undergoes large variations in appearance and structure. The proposed tracker contains two modules that require online updates: the appearance model (discriminative SVM classifier and generative color histogram), and the dynamic target graph $\mathcal{G}(V, E)$.

The discriminative SVM classifier is updated with the online SVM learning algorithm, *LASVM*, of [4]. The training samples (the color features of superpixels) are collected periodically. Those samples labeled as positive by graph cut segmentation in the target bounding box are trained as positive. The remaining ones are considered as negative. To avoid tracking drift problem caused by bad updating samples, samples are collected from both the initial and current frames. The generative foreground/background RGB histograms are also updated

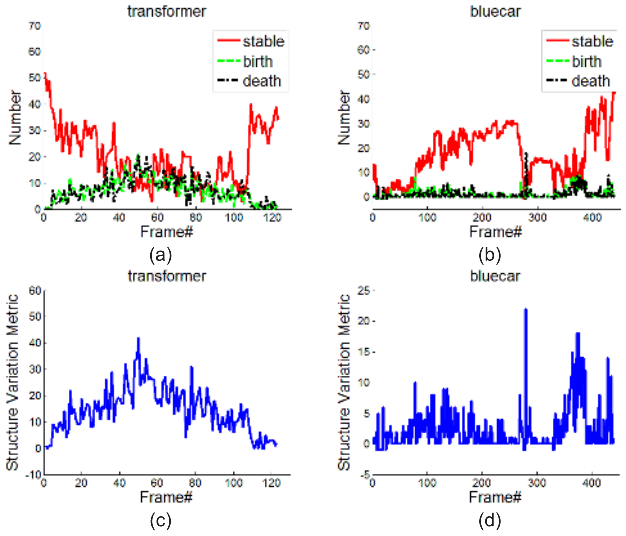


Fig. 4. (a) and (b) show the node variations in the three states of sequence *bluecar* and *transformer*, respectively. (c) and (d) are the illustrations of the structure variation metric over time of sequence *bluecar* and *transformer*, respectively. When y-coordinate is below 0, it means the target graph does not update.

incrementally, using

$$\begin{aligned}\mathcal{H}^f &= \mathcal{H}^f_{init} + \mathcal{H}^f_{hist} + \mathcal{H}^f_{curr}\\ \mathcal{H}^b &= \mathcal{H}^b_{init} + \mathcal{H}^b_{hist} + \mathcal{H}^b_{curr}\end{aligned} \tag{23}$$

where $\mathcal{H}^{(\cdot)}_{init}$, $\mathcal{H}^{(\cdot)}_{hist}$ and $\mathcal{H}^{(\cdot)}_{curr}$ are histograms derived from the initial frame, all previous frames, and current frame, respectively. This appearance updating mechanism not only keeps the initial information but also adapts to appearance variations.

To update the dynamic graph, we define three node states: **birth**, **stable** and **death**. The definitions are as follows.

- **birth:** a node i (a candidate local part in the final target bounding box) is in the birth state if 1) it cannot be matched to any node of $\mathcal{G}(V, E)$, and 2) its geometric distance to any other node satisfies $d_{i,j} > r\theta_b$, $\forall j \in \mathcal{G}(V, E)$, where $\theta_b = 0.7$. This distance constraint prevents the updated $\mathcal{G}(V, E)$ from being too dense.
- **stable:** a node i is in the stable state if it successfully matches some other node i'. A successful match occurs when $D(f_i, f_{i'}) > \theta_a$ and $||R_i - R_{i'}||_2 < r\theta_r$, where $\theta_a = 0.4$ and $\theta_r = 0.5$.
- **death:** a node is in the death state if it has not been successfully matched continuously for N_f or more frames.

After the target bounding box is determined, **death** nodes are deleted, **stable** nodes preserved, and **birth** nodes added to the target graph $\mathcal{G}(V, E)$. New edges are then introduced according to the geometric relations between nodes. Fig. 4(a) and 4(b) show the variation of nodes in the three states across two video sequences.

Model updates are conditioned on two constraints. The first, which follows from (22), is that $\gamma \cdot N^{mat}(l_c + \mu^*, s^*) + N^{pos}(l_c + \mu^*, s^*) - N^{neg}(l_c + \mu^*, s^*) < \theta_c$. The second one is that $N_{out} < \theta_s N_{in}$, where N_{in} and N_{out} are the numbers of candidate parts in and outside of the target bounding box, respectively. θ_c and θ_s are manually set thresholds.

```
**Algorithm 1** Proposed Tracking Algorithm
**Initialization:**
1. Initialize the SVM classifier and the target color histograms $\mathcal{H}^f$ and $\mathcal{H}^b$.
2. Initialize the target graph $\mathcal{G}(V, E)$.
**Tracking:**
**while** run **do**
    1. Oversegment the tracking window into a set of superpixels $\{T_p\}$
    2. Separate the candidate local parts $\{T_{i'}\}_{i'=1}^{n_Q}$ from the background with (7), and construct the candidate graph $\mathcal{G}'(V', E')$ with these candidate local parts and their interactions.
    3. Construct the affinity matrix $\mathbf{A}$, and do Eigendecomposition on $\mathbf{A}$ to get its leading eigenvector $\mathbf{x}^*$.
    4. Perform discretization on $\mathbf{x}^*$ to obtain $\mathbf{z}^*$ based on the constraints in (1), then get the optimal correspondences.
    5. Locate the optimal target state with (21), and fine-tune it with (22).
    6. Update the appearance model and the dynamic target graph $\mathcal{G}(V, E)$ when the updating conditions are satisfied.
**end while**
```

These conditions tend not to be met when the object undergoes severe occlusion or there is substantial noise from the background. In this case, no update takes place. Otherwise, as shown in Fig. 4, the target is usually updated. This updating mechanism enables the robust adaption of the target structure over time, as illustrated in Fig. 5. The details of the proposed tracker are described in Algorithm 1.

VIII. Experiments

In this section, two experiments are performed. In the first one (Section VIII-A), we design a metric for the degree of structure variation of the target, based on the change of the nodes in the dynamic target graph. In the second (Section VIII-B, VIII-C and VIII-D), we evaluate the performance of the proposed Dynamic Graph Tracker (DGT) and compare it to state-of-the-art results.

A. Structure Variation Metric

To the best of our knowledge, no metric of geometric structure variation of the target is available. This is because the target is usually represented by a bounding box and the bounding box cannot tell the geometric structure variation of the target very well. On the contrary, the graph representation enables a characterization of structure variation by counting nodes (parts). Based on our observation, the change of the nodes always coincides with the structure variation of the target, so in particular we measure the amount of structure variation of the target by

$$Mc^t = n_a^t + n_d^t$$

where n_a^t and n_d^t are the numbers of added and deleted nodes at time t, as described in Section VII. This metric is evaluated on *bluecar* and *transformer* sequences.

As shown in Fig. 4(c) and 4(d), the transformer object undergoes large structure variation throughout the whole sequence. Although there are some variations in the first 30 frames, it is not drastic. However, from ♯030 to ♯095, the structure changes severely. The transformation continues between ♯095 and ♯105, but is again less severe. Finally there is little variation in the last 15 frames. The sequence *bluecar*

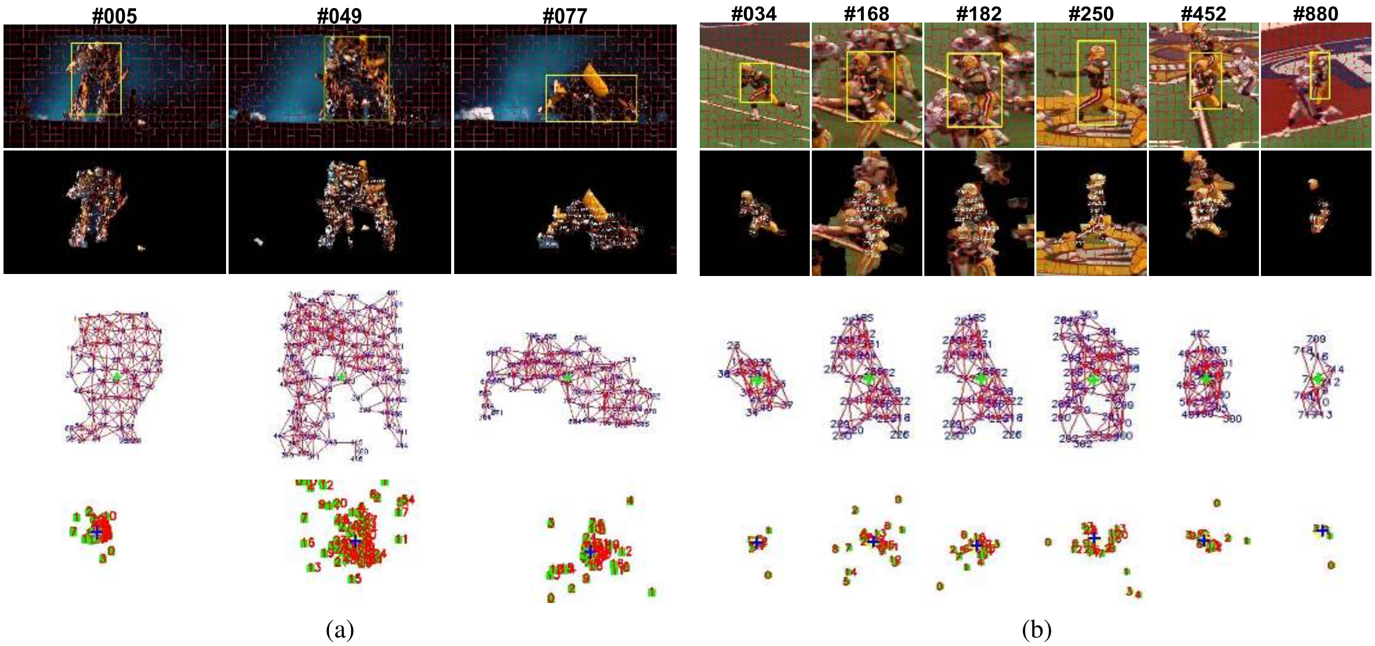


Fig. 5. The first row is the results of superpixel segmentation and the tracked target. The second row is the results of foreground/background separation in which the numbers on the parts mean the indices of the corresponding nodes in the target graph. The third row is the dynamic target graph where the numbers are the indices of nodes and the red line represents interaction between two neighboring nodes. The last row shows the weighted voting, where the green square is the voted center of every single correspondence, the number indicates the reliability degree of correspondence from small to big, the blue cross is the voted center without fine-tuning, and the yellow cross is the final target center with segmentation fine-tuning. The relative distance between the voted centers in the last row is enlarged for clear display. The practical voted centers are much more compact. (a) Transformer. (b) Football.

TABLE I

COMPARISON OF ACEP CRITERIA

Seq.	MIL[3]	IVT[23]	TLD[18]	$\ell 1$[29]	VTD[21]	CT[45]	Frag[2]	HABT[34]	BHMC[20]	SPT[40]	DGT-0	DGT-S
lemming [33]	14.9	128	167	140	92.4	19.7	82.8	107	158	**7.15**	13.4	7.51
waterski [17]	17.1	34.1	20.8	42.6	46.0	13.4	78.5	16.0	116	9.57	10.5	**6.95**
lipinski [17]	33.8	90.8	109	46.6	16.5	23.8	50.5	30.9	14.1	12.3	11.1	**7.22**
yunakim [17]	59.4	142	39.4	70.6	28.2	46.6	50.4	27.0	-	**16.8**	24.5	17.6
football [32]	102	236	197	146	118	127	75.8	33.1	30.2	179	11.7	**8.38**
kwan [32]	15.6	48.0	96.5	48.0	34.3	41.7	13.1	9.21	42.3	15.3	15.4	**6.87**
transformer [20]	47.7	139	25.5	269	42.9	29.2	36.6	141	23.2	**10.1**	19.6	10.2
gymnastics [20]	10.7	76.2	17.4	34.8	10.0	56.1	9.82	15.9	10.5	19.46	15.7	**6.41**
diving [20]	95.6	82.1	102	93.6	65.1	46.7	74.1	67.8	**10.6**	67.6	11.5	11.7
bolt [40]	356	360	-	48.3	16.3	189	241	126	122	**6.74**	11.6	7.38
basketball [21]	93.3	95.4	158	209	**7.64**	109	12.7	183	-	22.4	16.0	9.81
dancer [34]	16.3	15.1	13.3	10.3	10.8	14.9	19.3	19.1	18.9	**6.62**	14.8	7.14
seqD [19]	17.5	8.02	65.1	7.78	4.53	8.83	**4.22**	8.27	22.52	20.0	12.2	5.12
seqH [19]	5.27	1.77	4.93	1.31	1.22	6.22	**1.20**	75.2	11.4	9.69	3.78	7.10
bluecar	130	83.3	48.1	90.6	15.2	99.6	92.2	80.8	186	20.1	14.5	**9.86**
avatar	107	163	162	261	113	161	139	125	18.3	18.3	23.4	**12.1**
up	150	57.3	34.0	59.0	20.0	23.2	149	66.7	55.0	37.7	13.5	**7.91**
neymar	214	203	25.8	169	8.13	13.6	-	106	-	6.08	13.0	**5.88**

has milder structure variation than *transformer*. In fact, it changes slightly in the first 270 frames. There is, however, a sharp transformation at ♯280, followed by a period of little transformation between ♯285 to ♯345, and a new drastic variation from ♯345 to ♯390. In the end, the bluecar goes through a second period of no transformation. The joint observation of the metric in Fig. 4 and the video sequences shows that this metric can precisely capture the dynamic structure variation of the target.

B. Experiment Setup

To evaluate the tracking performance, we use a set of 18 challenging video sequences, 14 from prior works [17], [19]–[21], [32]–[34], [40], and the last 4 from our own collection. Altogether, the challenges in these sequences include structure deformation, severe occlusion, complex background, large scale changes and abrupt movements. The proposed DGT is compared to bounding box based trackers, including MIL [3], IVT [23], TLD [18], $\ell 1$ [29], VTD [21] and CT [45], and part based trackers, including Frag [2], HABT [34], BHMC [20] and SPT [40], with the results presented in Table I, and Table II. An illustration of the results is also given in Fig. 7. More results and the code can be found on our website.[2]

1) Implementation Details: Our tracker is implemented in C++ code and runs at approximately 4-10 frames per second, on a PC with 2.4 GHz CPU and 3 GB memory. For SLIC algorithm [1], the compactness is set to 50. The number

[2] https://sites.google.com/site/zhaoweicai1989/

TABLE II

COMPARISON OF SUCCESS RATE CRITERIA

Seq.	Frames	MIL[3]	IVT[23]	TLD[18]	ℓ1[29]	VTD[21]	CT[45]	Frag[2]	HABT[34]	BHMC[20]	SPT[40]	DGT-0	DGT-S
lemming	**1336**	0.8321	0.2127	0.1754	0.0970	0.3806	0.8246	0.5485	0.3806	0.0896	**0.9328**	0.8321	0.9179
waterski	**95**	0.6000	0.6947	0.6526	0.6105	0.5579	0.6842	0.4632	0.7368	0.0526	**0.8947**	0.8105	0.8842
lipinski	**660**	0.4697	0.0530	0.3182	0.2045	0.5530	0.2424	0.3409	0.1667	0.2879	0.1364	0.8258	**0.8712**
yunakim	**571**	0.1304	0.0434	0.1130	0.0261	0.1130	0.0252	0.0957	0.3609	-	0.5130	0.5304	**0.7826**
football	**958**	0.1719	0.0156	0.0365	0.1927	0.1771	0.1458	0.1882	0.2135	0.2473	0.2240	0.7083	**0.9427**
kwan	**751**	0.6623	0.2252	0.1788	0.3642	0.3642	0.3444	0.7947	0.8940	0.0397	0.3179	0.7219	**0.9669**
transformer	**124**	0.3871	0.3952	0.4032	0.3065	0.4032	0.4677	0.4032	0.2581	0.6290	**1.0000**	0.9839	**1.0000**
gymnastics	**763**	0.5882	0.2549	0.2288	0.0327	0.7320	0.2157	0.7386	0.7255	0.8431	0.2353	0.7843	**0.9477**
diving	**230**	0.1826	0.1435	0.1304	0.1174	0.1304	0.1957	0.2217	0.1348	**0.5455**	0.1522	0.2554	0.4348
bolt	**350**	0.0143	0.0143	-	0.3857	0.3429	0.0143	0.0143	0.0143	0.0143	0.7143	0.4143	**0.8857**
basketball	**725**	0.2414	0.1034	0.0207	0.1034	**0.9793**	0.2897	0.8690	0.1448	-	0.8069	0.4414	0.9310
dancer	**225**	0.8933	0.9422	0.7377	**0.9911**	**0.9911**	0.9822	0.8267	0.7644	0.6356	0.2311	0.8489	0.9822
seqD	**947**	0.6931	0.7345	0.5026	**0.9841**	0.9418	0.7460	0.9312	0.8634	0.0053	0.0635	0.3703	0.8677
seqH	**412**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.2561	0.4878	0.4756	1.0000	0.9756
bluecar	**441**	0.0449	0.1685	0.5169	0.5955	0.7978	0.0225	0.1798	0.2584	0.0225	0.3371	0.8989	**0.8539**
avatar	**134**	0.1119	0.0522	0.0970	0.1194	0.1269	0.1493	0.0522	0.1343	0.4254	0.3806	0.5373	**0.8955**
up	**190**	0.2000	0.5211	0.3158	0.4474	0.5526	0.2737	0.2789	0.1789	0.2053	0.3789	0.6684	**0.9368**
neymar	**311**	0.0794	0.0317	0.2540	0.0476	0.7778	0.4603	-	0.5714	-	**0.8889**	0.2698	0.6032
average	-	0.4057	0.3115	0.3342	0.3681	0.5512	0.3935	0.4675	0.3921	0.3021	0.4824	0.6612	**0.8711**

of superpixels varies according to the initial size of the target, ensuring the initial target includes approximately 15 to 30 superpixels. We usually set it between 200 to 500 in our experiments. For updating, the *LASVM* is updated every 3 frames, the target color histograms and graph model are updated every frame. N_f is in the interval [2,5], depending on the sequence, with lower values when the target structure changes more quickly. The updating thresholds are chosen as $\theta_c \in [0.5, 1.0]$ and $\theta_s \in [0.3, 0.6]$. The local perturbation of (22) is restricted to a window $\mu \in [-\phi, \phi] \times [-\phi, \phi]$, with ϕ equal to 4 or 8 in all experiments. ϕ could be slightly increased when segmentation performs well. The parameters of the other trackers are set to the default values suggested in the original papers or codes. The best one of 5 runs is chosen for comparison.

2) Evaluation Criteria: All trackers are evaluated under two criteria: Average Center Error in Pixels (ACEP) and Success Rate. Tracking is considered successful when it has a Pascal score higher than 0.5. The Pascal score is defined in the PASCAL VOC [11] as the overlap ratio between the tracked bounding box B_{tr} and ground truth bounding box B_{gt}: $\frac{area(B_{tr} \cap B_{gt})}{area(B_{tr} \cup B_{gt})}$. A lower ACEP and a higher Success Rate are indicative of better tracking performance.

C. Qualitative Analysis

1) Effective Dynamic Graph Representation: As shown in Fig. 5(a), the transformer undergoes large structure deformation, and it is not well captured by the traditional bounding box based appearance representation. Differently, the dynamic graph intuitively has the ability to represent the largely deformable target. In Fig. 5(b), the reader will find the target graphs at ♯168 and ♯182 are the same. This is because there is too much segmentation noise between these frames and the target graph is not updated. Although the dynamic graph does not represent the target very well in this period, the part correspondences are still satisfactory, since our undirected target graph is deformable and the pairwise geometric relations tolerate structure variation to some degree. The target graph also contains many outliers around ♯250. However, successful matching can still be obtained due to the robustness of the proposed affinity matrix.

2) Resistance to Noise: Since the inputs for graph matching are the separated candidate parts, the performance of the foreground/background separation is important. In reality however, the extracted candidate parts contain lots of noise caused by the complex background and the non-accurate foreground/background separation, as shown in Fig. 5(b). Moreover, the local parts always have great similarity with each other in color appearance, and the target graph also contains some outliers and noise. All of these make the matching and tracking difficult. By introducing several constraints to construct a less-noisy affinity matrix in Section V, the spectral matching method will find the optimal correspondences with less noise from very noisy inputs. For example, at ♯168, ♯182, ♯250, and ♯452 in Fig. 5(b), although many outliers exist, most matched candidate parts belong to the target and most outliers are not matched. The target center can be located robustly with these matched parts. On the contrary, we may not exactly know where the target is if the target location depends only on segmentation without correspondence results.

3) Reliability in Voting: In the absence of voting weights, incorrect correspondences would contribute equally to correct correspondences to the tracking process. The weighed voting procedure of Section VI allows reliable tracking in the presence of correspondence noise. This is visible in the last row of Fig. 5, where every green square represents the tracked center $l_c^{ii'}$ of (18) for every single correspondence. Note that these tracked centers are close to the real target center when the matching is good, as in ♯005 of *transformer* sequence, and ♯034 and ♯452 of *football* sequence. On the other hand, they tend to scatter when matching is poor, as in ♯049 of *transformer* sequence, and ♯168 and ♯250 of *football* sequence. The closer the tracked centers, the higher accuracy of the matching. The weighted voting mechanism of (19) suppresses the contribution of incorrect correspondences, while enhancing that of correct ones. This can be seen in Fig. 5, from the fact that the tracked centers $l_c^{ii'}$ of bad correspondences are usually far away from the target center with lower weights than good correspondences. This observation demonstrates the reliability of our weighted voting process.

D. Comparison Analysis

1) Comparison Among DGTs: We implement two versions of DGT to test the effectiveness of fine-tuning target location by μ^* of (22) using the foreground/background segmentation: one without fine-tuning (DGT-0) and the other one with fine-tuning (DGT-S). All of the other parameters for DGT-0 and DGT-S are the same in our experiments. Table I and II show that, while the performance of DGT-0 is quite good, the segmentation-based adjustment is beneficial. When the targets are large, e.g. in *lemming*, *lipinski*, *transformer* and *dancer*, the two trackers have similar performance. This is not surprising since, in this case, there are plenty of target parts to enable a robust graph matching. The gains of DGT-S by fine-tuning are more significant when targets are small, e.g. in *up*, *neymar* and *bolt*, since the matching and voting steps become less stable. When the targets rotate or undergo drastic deformation too quick, as in *yunakim*, *gymnastics*, *diving* and *avatar*, the dynamic graph representation cannot adapt to the variations immediately, thus the segmentation-based tuning is also helpful. On the other hand, DGT-0 has better performance over DGT-S in the sequence *bluecar*. This is mostly due to the fact that the bluecar target is occluded for a sizeable length of time (between ♯005 and ♯070), and only the unoccluded local parts are segmented from the background. If the tracker relies too much on segmentation, the dynamic graph will gradually shrink to the unoccluded region, limiting tracking accuracy in subsequent frames. If without fine-tuning, the dynamic graph will not shrink, and the accurate target center can still be located with the unoccluded parts.

Based on the good performance of DGT-0, it is convinced that graph matching is the foundation of the good performance of DGT-S. With more and more high-level computer vision tasks using segmentation, it is also quite reasonable to incorporate segmentation modification here and the performance is actually improved. The combination of these two parts is an attractive strategy for visual tracking.

2) Comparison to Other Trackers: The results of Tables I and II and Fig. 7 show that the DGT has superior performance to many state-of-the-art trackers in the literature. We next discuss the performance of these various trackers in response to different types of challenges.

a) Structure deformation: Structure deformation can be catastrophic for bounding box based trackers. For example, in the sequences *transformer*, *diving*, *yunakim* and *avatar*, where targets changes structure quickly and severely, the bounding box based trackers (IVT, MIL, TLD, $\ell 1$, VTD and CT) severely underperform part based trackers (Frag, HABT, BHMC, SPT, and DGT). This is because the part-based trackers focus on the appearance of local part, which is less sensitive to structure variation than bounding box based trackers. For the sequences with human motion, such as *waterski*, *lipinski*, *football*, *kwan*, *gymnastics*, *bolt*, *basketball*, *dancer*, *up* and *neymar*, while the bounding box representation can model the target well, the part-based trackers still own some advantages. On the remaining sequences, e.g. *lemming* and *bluecar*, where targets are rigid, the two representations have equivalent performance.

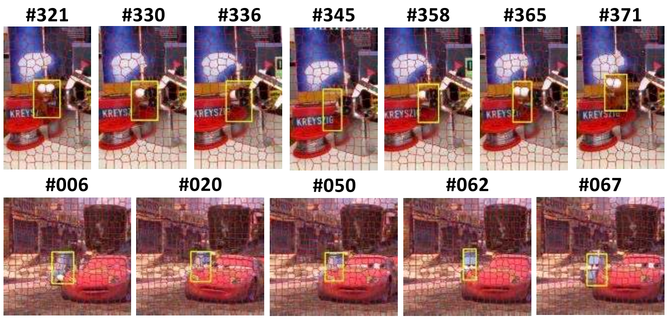


Fig. 6. Tracking results of DGT-0 under occlusion challenge.

Although part-based trackers own advantage in handling large structure deformation, the lack of effective updating and scaling does lead HABT and Frag to fail in the presence of large and fast structure deformations. On the other hand, the inability to stably track parts appears to be a major difficulty for BHMC, and SPT always shrinks to local region of the target due to a lack of global structure constraints. All these limitations are shown in Fig. 7. Differently, our DGT performs well in nearly all test sequences, since the inner structure of the target is exploited sufficiently and the dynamic graph effectively adapts to the structure deformation.

b) Occlusion: Sequences with non-trivial amounts of target occlusion, such as *lemming*, *basketball*, *football*, *neymar*, *bluecar* and *avatar*, create difficulties to methods, such as TLD, CT, and HABT, that do not have specific occlusion-handling mechanisms. On the contrary, $\ell 1$ can precisely find the target in *bluecar* where the target car is severely occluded, as demonstrated in Table I and Table II. Several methods, such as MIL, VTD and Frag, claim to be robust under occlusion, but they do not always have good performances in these sequences. This could be due to the fact that occlusion is combined with other challenges, such as large deformation and complex background, that these methods are not equipped to handle. Finally, in the absence of global constraints, part based trackers, such as BHMC and SPT, will shrink to the unoccluded parts of the target.

Although DGT-S will also shrink to unoccluded parts for sequences with severe and long periods of occlusion, DGT-0 performs very well in this case with the help of the visible local parts. As depicted in Fig. 6, DGT-0 can still find out the accurate bounding box of the target even when a few parts of the target are exposed. Only a small number of visible parts within the graph are enough for our tracker to recognize the whole graph and the whole target.

c) Illumination variations: Illumination variation has serious influence on appearance features. As shown in Fig. 7, the frequent illumination variations in the sequence *up* lead other trackers to drift away quickly, such as SPT, HABT, $\ell 1$, Frag, TLD, etc. On the other hand, the incremental subspace learning enables IVT to recognize the girl even when the sunlight is blocked by the balloons for several times. The online updates of *LASVM* classifier and color histograms

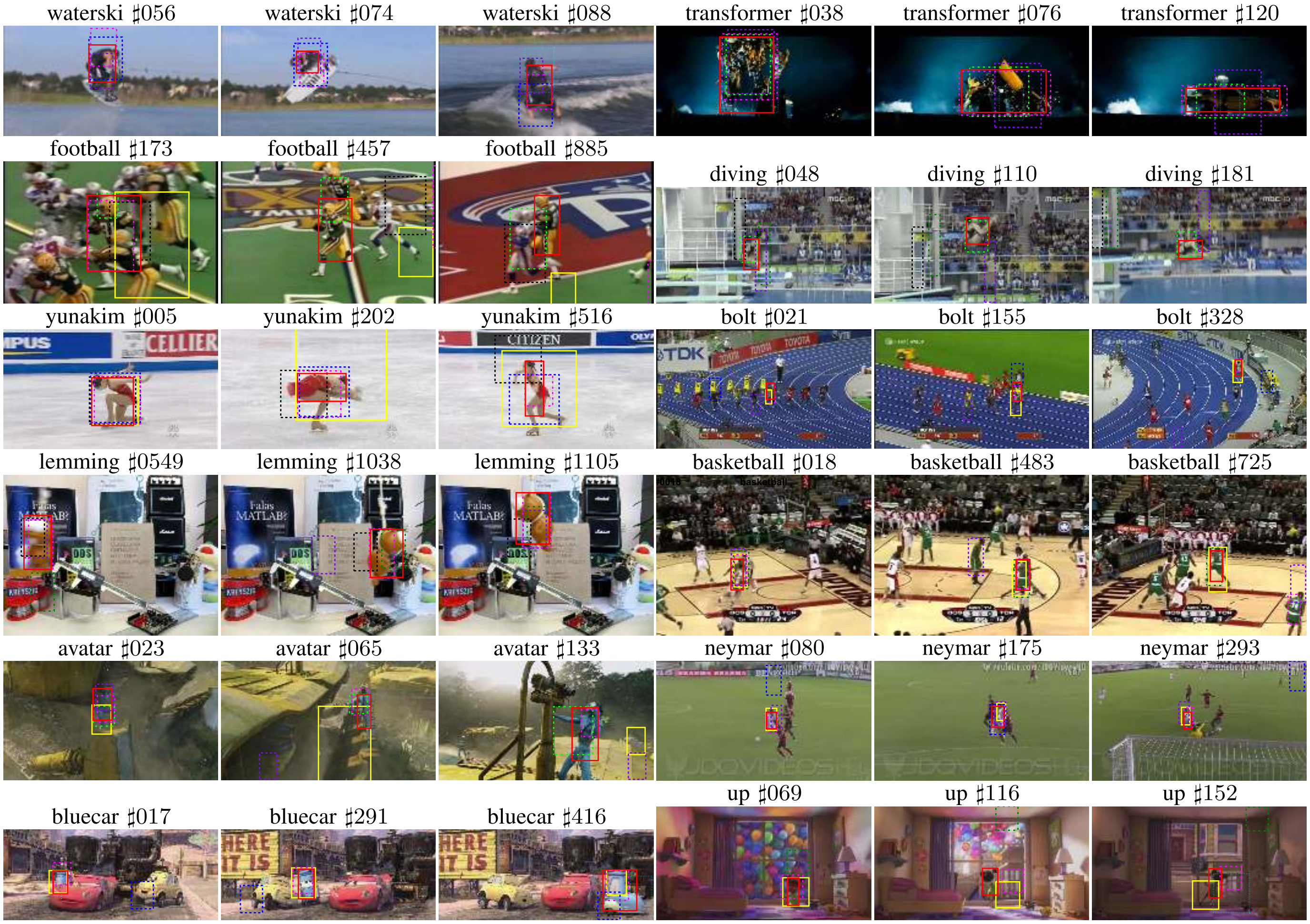


Fig. 7. Tracking results. The results of our DGT, MIL, TLD, VTD, CT, Frag, HABT, BHMC and SPT are depicted as red, black, cyan, yellow, purple, dark green, blue, light green, and magenta rectangles respectively. Only the trackers with relatively better performance of each sequence are displayed.

with initial and newly coming samples, provide DGT with resistance to the frequent illumination variations. Besides, even some parts of the target have severe appearance changes, the structure information will still help to recognize those corrupted parts.

d) Abnormal movement: Abnormal movements, such as fast motion (*avatar* and *bolt*), abrupt motion (*up*), and rotation (*yunakim*, *diving lipinski* and *waterski*), are always a great challenge for tracking, because these abnormal movements do not obey the movement assumption. For example, in *bolt*, many trackers lose the target when it begins to speed up. Similarly, VTD suddenly fails to track the target that jumps up abruptly in *up*. Most trackers are also quite unstable when faced with a fast spinning athlete in *yunakim*. The integration of appearance based segmentation and geometric constraints makes DGT much less sensitive to these types of abnormal movements.

IX. Conclusion

In this work, we have proposed the DGT to handle deformation and occlusion challenges, a noval dynamic graph based tracker that captures the inner structure information of the target by modeling the interactions between local parts. Target tracking is interpreted as matching the candidate graph to the target graph, and the matching is solved with recourse to spectral technique. Target location is determined by a set of weighted votes of matched parts according to their correspondences reliability, and refined by a foreground/background segmentation. Extensive experiments have shown that the DGT has tracking performance superior to many state of the art trackers, in the presence of various challenges, especially deformation and occlusion.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[2] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 1. Jun. 2006, pp. 798–805.

[3] B. Babenko, M.-H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[4] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, “Fast kernel classifiers with online and active learning,” *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, Dec. 2005.

[5] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[6] Z. Cai, L. Wen, J. Yang, Z. Lei, and S. Z. Li, “Structured visual tracking with dynamic graph,” in *Proc. 11th ACCV*, 2012, pp. 86–97.

[7] L. Cehovin, M. Kristan, and A. Leonardis, “An adaptive coupled-layer visual model for robust visual tracking,” in *Proc. IEEE ICCV*, Nov. 2011, pp. 1363–1370.

[8] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proc. IEEE CVPR*, vol. 2. Jun. 2000, pp. 142–149.

[9] T. Cour, P. Srinivasan, and J. Shi, “Balanced graph matching,” in *Advances in Neural Information Processing Systems 1*. Cambridge, MA, USA: MIT Press, pp. 313–320, 2006.

[10] A. Egozi, Y. Keller, and H. Guterman, “A probabilistic approach to spectral graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 18–27, Jan. 2013.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[13] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, “Bottom-up segmentation for top-down detection,” in *Proc. IEEE CVPR*, Jun. 2013, pp. 3294–3301.

[14] H. Grabner and H. Bischof, “On-line boosting and vision,” in *Proc. IEEE CVPR*, vol. 1. Jun. 2006, pp. 260–267.

[15] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, “Tracking the invisible: Learning where the object might be,” in *Proc. IEEE CVPR*, Jun. 2010, pp. 1285–1292.

[16] A. B. V. Graciano, R. M. Cesar, Jr., and I. Bloch, “Graph-based object tracking using structural pattern recognition,” in *Proc. SIBGRAPI*, Oct. 2007, pp. 179–186.

[17] M. Grundmann, V. Kwatra, M. Han, and I. Essa, “Efficient hierarchical graph-based video segmentation,” in *Proc. IEEE CVPR*, Jun. 2010, pp. 2141–2148.

[18] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N learning: Bootstrapping binary classifiers by structural constraints,” in *Proc. IEEE CVPR*, Jun. 2010, pp. 49–56.

[19] D. A. Klein and A. B. Cremers, “Boosting scalable gradient features for adaptive real-time tracking,” in *Proc. IEEE ICRA*, May 2011, pp. 4411–4416.

[20] 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 CVPR*, Jun. 2009, pp. 1208–1215.

[21] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Proc. IEEE CVPR*, Jun. 2010, pp. 1269–1276.

[22] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *Proc. 10th ICCV*, Oct. 2005, pp. 1482–1489.

[23] J. Lim, D. A. Ross, R.-S. Lin, and M.-H. Yang, “Incremental learning for visual tracking,” in *Advances in Neural Information Processing Systems 17*. Cambridge, MA, USA: MIT Press, 2004.

[24] B. Liu, J. Huang, L. Yang, and C. Kulikowski, “Robust tracking using local sparse appearance model and K-selection,” in *Proc. IEEE CVPR*, Jun. 2011, pp. 1313–1320.

[25] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[26] L. Lu and G. D. Hager, “A nonparametric treatment for location/segmentation based visual tracking,” in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[27] J. Maciel and J. Costeira, “A global solution to sparse correspondence problems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 187–199, Feb. 2003.

[28] V. Mahadevan and N. Vasconcelos, “Biologically inspired object tracking using center-surround saliency mechanisms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 541–554, Mar. 2013.

[29] X. Mei and H. Ling, “Robust visual tracking using $\ell 1$ minimization,” in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 1436–1443.

[30] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *Proc. 7th ECCV*, vol. 1. 2002, pp. 661–675.

[31] A. Quattoni, M. Collins, and T. Darrell, “Conditional random fields for object recognition,” in *Advances in Neural Information Processing Systems 17*. Cambridge, MA, USA: MIT Press, 2004.

[32] X. Ren and J. Malik, “Tracking as repeated figure/ground segmentation,” in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[33] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, “PROST: Parallel robust online simple tracking,” in *Proc. IEEE CVPR*, Jun. 2010, pp. 723–730.

[34] S. M. S. Nejhum, J. Ho, and M.-H. Yang, “Online visual tracking with histograms and articulating blocks,” *Comput. Vis. Image Understand.*, vol. 114, no. 8, pp. 901–914, 2010.

[35] R. Szeliski *et al.*, “A comparative study of energy minimization methods for Markov random fields with smoothness-based priors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008.

[36] F. Tang and H. Tao, “Probabilistic object tracking with dynamic attributed relational feature graph,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1064–1074, Aug. 2008.

[37] M. Tian, W. Zhang, and F. Liu, “On-line ensemble SVM for robust object tracking,” in *Proc. 8th ACCV*, vol. 1. 2007, pp. 355–364.

[38] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, “Segmentation as selective search for object recognition,” in *Proc. IEEE ICCV*, Nov. 2011, pp. 1879–1886.

[39] U. von Luxburg, “A tutorial on spectral clustering,” *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[40] S. Wang, H. Lu, F. Yang, and M.-H. Yang, “Superpixel tracking,” in *Proc. IEEE ICCV*, Nov. 2011, pp. 1323–1330.

[41] W. Wang and R. Nevatia, “Robust object tracking using constellation model with superpixel,” in *Proc. 11th ACCV*, vol. 3. 2012, pp. 191–204.

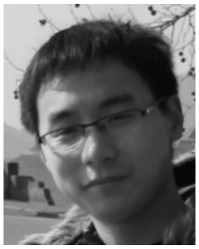
[42] L. Wen, Z. Cai, Z. Lei, D. Yi, and S. Z. Li, “Online spatio-temporal structural context learning for visual tracking,” in *Proc. 12th ECCV*, vol. 4. 2012, pp. 716–729.

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

[44] M. Yang, J. Yuan, and Y. Wu, “Spatial selection for attentional visual tracking,” in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[45] K. Zhang, L. Zhang, and M.-H. Yang, “Fast compressive tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.

Zhaowei Cai received the B.S. degree in automation from Dalian Maritime University, Dalian, China, in 2011. From 2011 to 2013, he was a Research Assistant with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA, USA. His research interests are in computer vision and machine learning.



Longyin Wen is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He received the B.S. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China, in 2010. His research interests are computer vision, pattern recognition, and in particular, object tracking.

Zhen Lei received the B.S. degree in automation from the University of Science and Technology of China, Hefei, China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010, where he is currently an Associate Professor. His research interests are in computer vision, pattern recognition, image processing, and in particular, face recognition. He has authored over 80 papers in international journals and conferences. He serves as an Area Chair of the International Joint Conference on Biometrics in 2014, the IAPR/IEEE International Conference on Biometric in 2015, and the IEEE International Conference on Automatic Face and Gesture Recognition in 2015.

Nuno Vasconcelos (S'92–M'00–SM'08) received the bachelor's degree in electrical engineering and computer science from the Universidade do Porto, Porto, Portugal, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the Department of Electrical and Computer Engineering, University of California at San Diego, San Diego, CA, USA, where he heads the Statistical Visual Computing Laboratory. He has received the National Science Foundation CAREER Award and the Hellman Fellowship Award. He has authored over 150 peer-reviewed publications.

Stan Z. Li (M'92–SM'99–F'09) received the B.Eng. degree from Hunan University, Changsha, China, the M.Eng. degree from the National University of Defense Technology, Changsha, and the Ph.D. degree from Surrey University, Surrey, U.K. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, where he is also the Director of the Center for Biometrics and Security Research, and the Center for Visual Internet of Things Research. He was with Microsoft Research Asia, Beijing, from 2000 to 2004, as a Researcher. Prior to that, he was an Associate Professor with Nanyang Technological University, Singapore. His research interest includes pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance. He has authored over 200 papers in international journals and conferences, and authored and edited eight books. He was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the Editor-in-Chief of the *Encyclopedia of Biometrics*. He served as the Program Co-Chair of the International Conference on Biometrics in 2007 and 2009, the International Joint Conference on Biometrics in 2014, and the 11th IEEE Conference on Automatic Face and Gesture Recognition, and the General Chair of the 9th IEEE Conference on Automatic Face and Gesture Recognition. He has been involved in organizing other international conferences and workshops in the fields of his research interest.