Real-time part-based visual tracking via adaptive correlation filters

Ting Liu, Gang Wang
School of Electrical and Electronic Engineering,
Nanyang Technological University.
{liut0016, wanggang}@ntu.edu.sg

Qingxiong Yang
Department of Computer Science,
City University of Hong Kong.
qiyang@cityu.edu.hk

Abstract

Robust object tracking is a challenging task in computer vision. To better solve the partial occlusion issue, part-based methods are widely used in visual object trackers. However, due to the complicated online training and updating process, most of these part-based trackers cannot run in real-time. Correlation filters have been used in tracking tasks recently because of the high efficiency. However, the conventional correlation filter based trackers cannot deal with occlusion. Furthermore, most correlation filter based trackers fix the scale and rotation of the target which makes the trackers unreliable in long-term tracking tasks. In this paper, we propose a novel tracking method which track objects based on parts with multiple correlation filters. Our method can run in real-time. Additionally, the Bayesian inference framework and a structural constraint mask are adopted to enable our tracker to be robust to various appearance changes. Extensive experiments have been done to prove the effectiveness of our method.

1. Introduction

Visual tracking is an important technique in computer vision with various applications such as security and surveillance, human computer interaction and auto-control systems [25, 43, 50]. With the development of single object tracking methods, most of the tracking tasks in simple environment with slow motion and slight occlusion can be solved well by current algorithms. However, in more complicated situations, more robust tracking methods are required to realize accurate and real-time tracking.

Current tracking algorithms are classified as either generative or discriminative methods. Generative methods treat tracking problem as searching for the regions which are the most similar to the tracked targets [3, 4, 6, 22, 24, 26–28, 46, 52]. The targets are often represented by a set of basis vectors from a subspace (or a series of templates). Different from generative trackers, discriminative approaches cast tracking as a classification problem that distinguishes the tracked targets from the backgrounds [2, 29, 42, 56]. It employs the information from both the target and background. For example, Avidan [3] proposed a strong classifier based on a set of weak classifiers to do ensemble tracking. In [4], Babenko et al. used an online multiple instance learning which puts all ambiguous positive and negative samples into bags to learn a discriminative model for tracking. Kalal et al. [21] proposed a P-N learning algorithm to learn tracking classifiers from positive and negative samples. In [16], Hare et al. used an online structured output support vector machine to adaptively track the targets. This category of approach is termed tracking-by-detection. Recently, correlation filter based tracking-by-detection methods have been proven to be able to achieve fairly high speed and robust tracking performance in relatively simple environments [7, 17].

Recently part-based tracking methods [1, 9, 19, 34, 37, 47] become more popular partially because of their favorable property of robustness against partial occlusion. They model the object appearance based on multiple parts of the target. Obviously, when the target is partially occluded, remaining visible parts can still provide reliable cues for tracking. Most of these methods can be viewed as tracking by part-based object matching over time in a video sequence. Because computational complexity of these methods is high, it is difficult to realize real-time tracking.

In this paper, We aim to build a real-time part-based visual object tracker which is able to handle partial occlusion and other challenging factors. Our key idea is to adopt the correlation filters as part classifiers. Hence, the part evaluation speed can be fast. However, it becomes very critical to combine the tracking scores of different parts in a proper way. If some parts are occluded, and we still assign big weights to them, then there will be tracking errors. Our contribution is to develop new criteria to measure the performance of different parts, and assign proper weights to them. Specifically, we propose to use Smooth Constraint of Confidence Maps as a criterion to measure how likely a part is occluded. The experiments proved the effectiveness of the SCCM criterion. Besides, we developed the s-

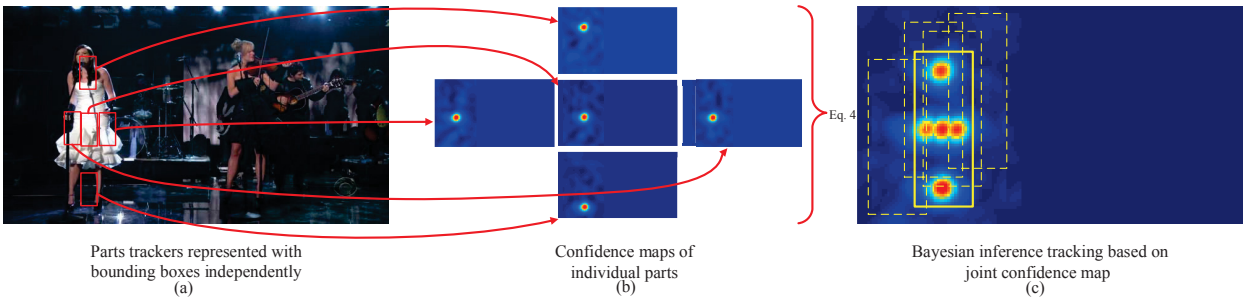


Figure 1: Each part tracker independently tracks the corresponding part and outputs a response map. The separate response maps are combined by Eq. 4. We track the whole target based on the joint confidence map in the Bayesian inference framework. When computing the likelihood, sampling candidates are shown as yellow rectangles. The solid yellow rectangle shows the tracking result on the confidence map with the maximum likelihood.

patial layout constraint method to: 1) effectively suppress the noise caused by combining of individual parts, 2) estimate the correct size of bounding box when the target is occluded.

2. Related work

Correlation filter based tracking methods

Correlation filter based trackers have been proven to be competitive with far more complicated tracking methods [7, 12, 17, 33]. In the correlation filter based trackers, the target is initially selected based on a small tracking window centered on the object in the first frame. The target is tracked by correlating the filter over a larger search window in next frame; the location with the maximum value in the correlation response indicates the new location of the target. The filter (also called classifier when it is used for tracking) is updated online based on the new location.

Particle filter based tracking methods Visual tracking can be considered as an estimation of the state for a time series state space model [8, 40]. The problem can be formulated in probabilistic terms. Particle filtering is a technique for implementing a Bayesian inference filter by Monte Carlo simulation, which is popular in visual tracking. It recursively constructs the posterior probability density function of the state space using Monte Carlo integration. Because it is computationally efficient and insensitive to local minima, particle filters have been widely used as tracking framework. Based on the particle filter framework, Ross et al. [34] incrementally learned the low dimensional subspace representation for target to adapt to the changes of target appearance. In [31], a series of target trivial templates were used to model the tracked target with the sparsity constraints. Jia et al. [20] proposed a structural local sparse coding model for the Bayesian inference based tracking method. In our method, the joint confidence maps are used in the Bayesian inference framework to infer the candidate with maximum posterior probability. To better enforce the spatial layout of parts, a structural constraint mask is adopted to calculate the likelihood of the observation and state.

Object tracking with parts

To handle the occlusion, many trackers divide the entire targets into separate parts [10, 11, 15, 18, 30, 38, 39, 41, 45, 48, 51]. In [36], the foreground shape was modelled as a small number of rectangular blocks. The algorithm tracked objects by matching the intensity histograms, and updated the shape by adjusting small blocks of the tracked window. Kwon et al. [23] presented a local patch-based appearance model which was updated online. In [54], Zhang et al. tracked targets by parts matching among multiple frames. Another work similar to ours is [49], in which the part-based tracking problem was solved by latent structured learning. However, the computational complexity of these methods is high; consequently, it is difficult for multiple part based trackers to run in real-time.

3. Part-based tracker via correlation filter

We aim to build a real-time part-based tracking system which is robust to occlusion and deformation. As introduced above, to realize efficient tracking in the real world, complicated tracking systems cannot be extended to multiple parts, since the computational complexity is high. Recently, the tracking system with the Kernelized Correlation Filter (KCF) [17] achieves very good performance with high speed (360 frames per second). Due to the competitive performance and efficiency, we develop our method based on the KCF tracker. Our key idea is to employ the fast KCF classifier as the part classifier, and develop a new method to

adaptively combine the part classification scores and adaptively update the model. Furthermore, to effectively solve the scale, rotation, skew, etc. problems for real-time tracker, a Bayesian inference framework is employed to achieve more robust performance.

3.1. The KCF tracker

The Convolution Theorem states that in the Fourier domain, the convolution of two patches can be computed by element-wise product, which is much more efficient. Hence, for the correlation filter based trackers, correlation is computed in the Fourier domain through Fast Fourier Transform (FFT); and the correlation response can be transformed back into the spatial domain using the inverse FFT. Although the correlation filter based trackers have high efficiency, because they model the holistic appearance of the target to train and update their models, the existing correlation filter based trackers cannot handle the occlusion problem well. Once the targets are occluded heavily, the trackers may fail to relocate the objects.

In this section, we briefly introduce the KCF tracking method. Readers may refer to [17] for more details. The classifier of KCF is trained using an image patch x of size $\mathrm{W} \times \mathrm{H}$. The training image patch is centred around the target. Taking advantages of the cyclic property and appropriate padding, KCF considers all cyclic shifts $x_{\mathrm{w,h}}$, $(\mathrm{w}, \mathrm{h}) \in \{0, ..., \mathrm{W}-1\} \times \{0, ..., \mathrm{H}-1\}$ as the training examples for the classifier. The regression targets y follow a Gaussian function, which takes a value of 1 for a centered target, and smoothly decays to 0 for any other shifts, ie. $y(\mathrm{w}, \mathrm{h})$ is the label of $x_{\mathrm{w,h}}$.

The goal of training is to find a function $f(z) = w^T z$ that minimizes the squared error over samples $x_{\mathrm{w,h}}$ and their regression targets $y(\mathrm{w}, \mathrm{h})$,

$$\min_{w} \sum_{\mathrm{w,h}} |\langle \phi(x_{\mathrm{w,h}}), w \rangle - y(\mathrm{w}, \mathrm{h})|^2 + \lambda \|w\|^2 \qquad (1)$$

where ϕ represents the mapping to the Hilbert space induced by the kernel κ. The inner product of x and x' is computed as $\langle \phi(x), \phi(x') \rangle = \kappa(x, x')$. λ is a parameter for the regularization term.

After mapping the inputs of a linear problem to a nonlinear feature-space $\phi(x)$, the solution w can be expressed as $w = \sum_{\mathrm{w,h}} \alpha(\mathrm{w}, \mathrm{h}) \phi(x_{\mathrm{w,h}})$.

$$\alpha = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(y)}{\mathcal{F}(k^x) + \lambda}\right) \qquad (2)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively; $(k^x) = \kappa(x_{\mathrm{w,h}}, x)$. Note that the vector α contains all the $\alpha(\mathrm{w}, \mathrm{h})$ coefficients. The target appearance $\hat{x}$ is learned over time. In the KCF tracker, the model consists of the learned target appearance $\hat{x}$ and the transformed classifier coefficients $\mathcal{F}(\alpha)$.

In the tracking, a patch z with the same size of x is cropped out in the new frame. The confidence score is calculated as

$$\hat{f}(z) = \mathcal{F}^{-1}(\mathcal{F}(k^z) \odot \mathcal{F}(\alpha)) \qquad (3)$$

where $\odot$ is the element-wise product; $(k^z) = \kappa(z_{\mathrm{w,h}}, \hat{x})$. $\hat{x}$ denotes the learned target appearance.

3.2. Adaptive weighting for part-based tracker

In visual tracking, partial occlusion is one of the main challenging factors that limit the performance. Intuitively, we can divide the target into small parts and track these parts independently. When some of the parts are occluded or deform, we can still locate the entire target correctly relying on the other parts. However, multi-part tracking is usually slow, because of the complicated training and updating processes. In this paper, we aim to develop a real-time part-based tracking system that is robust to occlusion and deformation.

The location of the target object is given in the 1st frame of the video, and the tracker is then required to track the object (by predicting a bounding box containing the object) from the 2nd frame to the end of the video. The target is divided into several parts. For each part of the object we run an independent KCF tracker that outputs a response map (confidence map). The response map is the correlation response used to locate the position of target part. The maps are then combined to form a single confidence map for the whole target that is used in the Bayesian inference framework.

The difficulty in this method is how to combine the confidence maps of different part trackers. In different frames, different parts of targets may suffer different appearance changes, illumination variation or occlusion. If we simply combine confidence maps with the same weight, the response of falsely tracked parts may be unfairly emphasized. Given the detection result of each part tracker, the contribution made for the joint confidence map should be different from each part, ie. the response of more reliable parts should be given larger weights. Through adaptively weighting each part response, the joint confidence map puts more emphasis on reliable parts and eliminates the clutters caused by drifting parts. For correlation filter based classifier, the peak-to-sidelobe ratio (PSR) (Eq. 6) can be used to quantify the sharpness of the correlation peak. The higher PSR value means more confident detection (in tracking system, it means the matching score between current frame and previous frames is high). Therefore, the PSR can be adopted to weight the confidence maps of parts under different situations. In addition, for tracking problems, the temporal smoothness property is helpful for detecting whether the target is occluded. Taking this observation into consideration and as validated from our experiments, we propose that

the smooth constraint of confidences maps should be considered for the weight parameters. As shown in Fig. 2, the responses of occluded parts should be combined with smaller weights. We define the smooth constraint of confidence maps (SCCM) in Eq. 7.

The joint confidence map at the t-th frame is defined as:

$$C^t = \sum_{i=1}^{N} \mathrm{w}_i^t \hat{f}_{\mathrm{p}(i)}^t, \tag{4}$$

where $\hat{f}_{\mathrm{p}(i)}^t$ is the confidence map (Eq. 3) of the i-th part at time t. $\mathrm{p}(i)$ denotes the relative position of part response in the joint confidence map C^t; it is determined by the maximum value of the part confidence map. N is the number of parts used to divide the target. w_i^t is the weight parameter of corresponding part.

$$\mathrm{w}_i^t = PSR_i + \eta \cdot \frac{1}{SCCM_i} \tag{5}$$

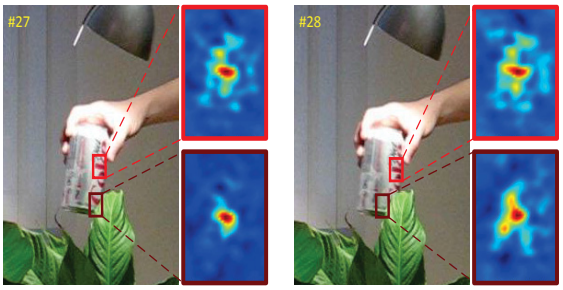
$$PSR_i = \frac{\max\left(\hat{f}_{\mathrm{p}(i)}^t\right) - \mu_i}{\sigma_i}. \tag{6}$$

$$SCCM_i = \left\| \hat{f}_{\mathrm{p}(i)}^t - \hat{f}_{\mathrm{p}(i)}^{t-1} \oplus \Delta \right\|_2^2 \tag{7}$$

where, μ_i and σ_i are the mean and the standard deviation of the i-th confidence map respectively. η is the trade-off between correlation sharpness and smoothness of confidence maps; in our experiments, it is simply set as 1. $\oplus$ means a shift operation of the confidence map, and Δ denotes the corresponding shift of maximum value in confidence maps from frame $t-1$ to t. $\hat{f}_{\mathrm{p}(i)}^{t-1}$ and $\hat{f}_{\mathrm{p}(i)}^t$ denote the individual response maps of part i, as shown at the right of Fig. 2(a) and 2(b). Because the location of part may shift in corresponding response maps, Δ is considered when calculating the smoothness of them in consecutive frames. As shown in Fig. 2(b), when the target part is occluded by the leaf, the PSR value is similar to that of the correctly tracked part. In this case, the term of SCCM is required to suppress the contribution of responses from the occluded parts.

3.3. Adaptive classifier updating

In tracking, the object appearance will change because of a number of factors such as illumination and pose changes. Hence it is necessary to update the part classifiers over time. In the KCF tracker, the model consists of the learned target appearance and the transformed classifier coefficients. They are computed by only taking the appearance of current appearance into account. The tracker then employs an ad-hoc method of updating the classifier coefficients by simple linear interpolation: $\mathcal{F}(\alpha)^t = (1-\gamma)\mathcal{F}(\alpha)^{t-1} + \gamma\mathcal{F}(\alpha)$, where $\mathcal{F}(\alpha)$ is the classifier coefficients and γ is a learning rate parameter. KCF used a fix learning rate, which means that the appearance and correlation filter will be updated



(a) (b)

Figure 2: Tracking target from frame #27 and #28, in which two parts are marked in bright and dark red bounding boxes respectively. The corresponding confidence maps of two part trackers are shown at the right with corresponding color bounding boxes. For the bright red part, it is tracked correctly between the consecutive frames. For the dark red part, it drifts in (b) because of the occlusion. To better evaluate the joint confidence map, the response of drifting part should contribute less. If we only consider PSR to weight the confidence maps, bright red and dark red parts will contribute equally. Different from the dark red part, the confidence maps of the bright red part in (a) and (b) have the smooth property. Hence, we should consider the smooth constraint to make weight selection more robust.

without adaptation to specific video frames. Once the tracker loses the object, the whole model will be contaminated in the remaining frames. We tackle this problem by adaptively updating the tracking model.

It is apparent that the model of an occluded part should not be updated to avoid introducing errors. Similar to the weight parameters discussed in section 3.2, a threshold is used to adaptively update each part tracker separately. The learning rate for the model is set proportional to the weight value (Eq. 5), ie. the part trackers with higher detection scores should be updated more because they are seen as more robust tracking parts. Therefore the update scheme is defined as:

$$\mathcal{F}(\alpha)_i^t = \begin{cases} (1-\beta\mathrm{w}_i^t)\mathcal{F}(\alpha)_i^{t-1} + \beta\mathrm{w}_i^t\mathcal{F}(\alpha)_i & if\ \mathrm{w}_i^t > \text{threshold} \\ \mathcal{F}(\alpha)_i^{t-1} & \text{else} \end{cases} \tag{8}$$

$$\hat{x}_i^t = \begin{cases} (1-\beta\mathrm{w}_i^t)\hat{x}_i^{t-1} + \beta\mathrm{w}_i^t x_i & if\ \mathrm{w}_i^t > \text{threshold} \\ \hat{x}_i^{t-1} & \text{else} \end{cases} \tag{9}$$

Contrary to traditional correlation filter based trackers, due

to the adaptive updating scheme, even when all the parts of target are occluded at one frame, our method can still maintain the accuracy of the classifier by using classifiers of previous frames. Hence, it is able to relocate the occluded target when it appears in the following frames.

For KCF tracker, when the target is tracked under normal conditions, the PSR typically ranges between 8.0 and 15.0, which indicates very strong peaks. We found that when PSR drops to around 5.0, it is an indication that the detection of corresponding part is not reliable. When the value of SCCM is larger than 0.5, it means that the parts may suffer heavy occlusion. We set the threshold in Eq. 8 and Eq. 9 as 7. The other learning rate β is fixed as 0.01.

Algorithm 1 Multiple part tracking

1: **Inputs:** t**-th frame** F_t**, the previous tracking state** s^{t-1};
2: Calculate the confidence map of each part;
3: Combine the confidence maps based on Eq. (4);
4: Apply the affine transformation on s^{t-1} to obtain a number of tracking states s_j^t;
5: Calculate the posterior probability $p(s_j^t|O^t)$ according to Eq. (11) and Eq. (13);
6: Predict the tracking state by $\hat{s}^t = \arg\max_{s_j^t} p(s_j^t|O^t)$;
7: **if** $\mathrm{w}_i^t >$ threshold Eq. (8, 9) **then**
8: Update the model;
9: **else**
10: Keep the model unchanged;
11: **end if**
12: **Output:**
13: **Find the location of each part** $\mathrm{p}(i)$**.**
14: **Find the target location** $\hat{s}_t$ **on confidence map and the corresponding position on the video frame;**

3.4. Tracking based on confidence map

Based on parts, a naive method to determine the position of the whole target is to compute the geometric center based on the tracking results of all the parts. However, due to the heavy occlusion and shape deformation, some of the part trackers may drift and the final bounding box may not be able to locate the target correctly. In [4, 16], dense sampling methods have been adopted to search for the state of the target objects. However, dense sampling requires high computational load and reduces the speed of the whole tracking system. In this paper, we carry out the tracking problem as a Bayesian inference task.

Let s^t denote the state variable describing the affine motion parameters of an object at the time t (e.g. location or motion parameters) and define $O^t = [o^1, o^2, \cdots, o^t]$ as a set of observations with respect to joint confidence maps. The optimal state $\hat{s}^t$ is computed by the maximum a posterior (MAP) estimation

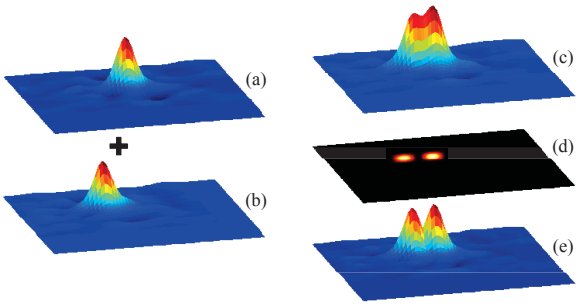


Figure 3: (a) and (b) are the response maps of two part trackers. (c) is the summation of (a) and (b). (d) is the structural constraint mask to enforce the spatial layout of individual parts. (e) is the confidence map used for state parameter searching after convolving with the mask.

$$\hat{s}^t = \arg\max_{s_j^t} p(s_j^t|O^t) \tag{10}$$

where s_j^t is the state of the j-th sample. The posterior probability is calculated recursively by the Bayesian theorem,

$$p(s^t|O^t) \propto p(o^t|s^t) \int p(s^t|s^{t-1}) p(s^{t-1}|O^{t-1}) ds^{t-1} \tag{11}$$

The $p(o^t|s^t)$ is the observation model.

The dynamics between states in this space is usually modelled by the Brownian motion. Each parameter in s^t is modelled independently by a Gaussian distribution given its counterpart in s^{t-1}.

$$p(s^t|s^{t-1}) = \mathcal{N}(s^t, s^{1:t-1}, \Psi) \tag{12}$$

where Ψ is a diagonal covariance matrix whose elements are the corresponding variances of affine parameters. The observation model $p(o^t|s^t)$ denotes the likelihood of the observation o^t at state s^t. Maximizing the posterior in Eq. 10 is equivalent to maximizing the likelihood $p(o^t|s^t)$. Thus, the likelihood calculation is one of the most important factors in our tracker.

In the traditional Bayesian inference based tracking methods, because more complicated features are adopted to represent the objects, a set of basis vectors or templates need to be built to calculate the likelihood. However, we are applying the Bayesian inference framework into confidence maps. We can simply calculate the sum of confidence scores in the candidate box as the likelihood value. In this process, we should enforce the spatial constraint: the summation between neighbouring part responses should not influence the likelihood calculation. Hence, we introduce a spatial layout

constraint mask as shown in Fig. 3(d). By applying this mask, we enforce the structural relationships among parts. In our method, the observation model is constructed by

$$p(o^t|s^t) = \frac{1}{|M^t|}\sum C^t(s^t) \odot M^t \qquad (13)$$

where $\odot$ is the element-wise production. M^t is a spatial layout constraint mask built by N cosine windows which gradually reduce the pixel values near the edge to zero. The relative positions of these cosine windows are determined by the maximum value of corresponding part response maps. The size of cosine window is determined by the size of tracking part. $|M^t|$ is the number of pixels within the mask. $C^t(s^t)$ means a candidate sample in the joint confidence map (the yellow rectangles in Fig. 1(c)). The summation is based on all the pixels within the candidate window. Besides the effect on spatial layout enforcement, by using such a structural constraint mask, we can make our method robust to the response noise of correlation filters.

The baseline method KCF used a fixed size of bounding box, which might lead to drifting when the scale of the object changed significantly. Benefiting from the multi-part scheme and Bayesian framework, our tracker is capable of solving the challenges of scale changes of targets. Fig. 4 shows one example of scale changes. Although the size of individual part trackers is fixed, the size of the whole target is determined through the Bayesian framework which is able to solve the challenges such as scale change, rotation and skew.

Ideally, the individual parts should stay close to each other to cover the entire target. However, in some challenging situations, such as significant appearance and illumination changes, some part trackers may move far away from the target. When this happens, we relocate the drifting parts using the correctly tracked parts based on spatial layout.

4. Experiments

To evaluate our proposed tracker, we compile a set of 16 challenging tracking sequences. These videos are recorded in indoor and outdoor environments and have variations of occlusion, illumination, pose changes , etc. We compared the proposed algorithm with thirteen state-of-the-art visual trackers: Frag[1], TLD [21], IVT [34], DFT [35], ORIA [44], LOT [32], CXT [13], L1apg [5], MTT [53], ASLA [20], MIL [4], Struck [16] and KCF [17]. All our experiments are performed using MATLAB R2012b on a 3.2 GHZ Intel Core i5 PC with 16 GB RAM. We use the source codes provided by the authors. The parameters of these trackers are adjusted to show the best tracking performance.

The size of part is selected by experimental validation on a number of sequences. We find when the size is between 1/4 and 1/6 of the object size,the results do not change much. However, when it is too big or too small, the results can be worse because a part should be at the right size to contain enough information while to be robust to occlusion. We also find the number of part doesn't affect the results much when it is larger than 5. We use 5 parts and 400 candidate samples.

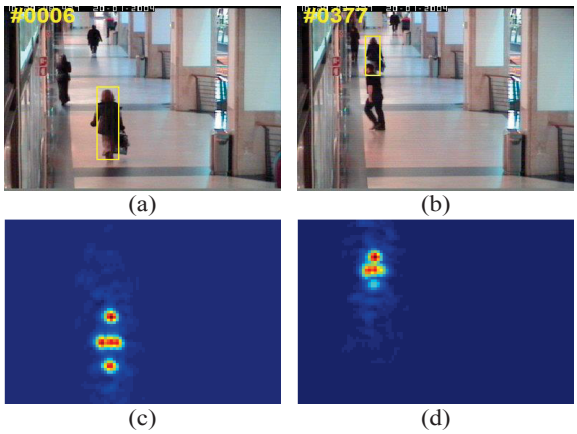


Figure 4: (a) The target is near the camera and caught with a larger bounding box. (b)The target is far away from the camera with a small bounding box. (c) The corresponding joint confidence map of (a). (d) The corresponding joint confidence map of (b). Because of the occlusion, the responses of bottom part are suppressed by the weighting method

4.1. Speed analysis

The major computational cost of the proposed method is KCF tracking for the N parts and candidates sampling. When using 5 parts and 400 sampled windows, the speed of the proposed method is 30 frames per second without code optimization. We compare the speed of our method with several part-based trackers [10, 15, 18, 36, 49, 55]. The running speed of trackers [15], [10] and [36] on the same hardware mentioned above is 10 FPS, 2 FPS and 3.7 FPS respectively. We didn't find the code of the other 3 trackers. To fairly compare with them, we have tested our method on the same hardware of [18] [55] (DualCore 2.7 GHz,4GB RAM) and [49] (Core2 2.6 GHz,4GB RAM); the average speed on them is 20 and 22 FPS respectively, which is faster than the corresponding ones. It can be observed that our algorithm is faster than all of these part-based trackers. More speed comparison with other state-of-the-art trackers is listed at the end of Table 1.

4.2. Quantitative evaluation

To assess the performance of the proposed tracker, two criteria, the center location error as well as the overlap rate,

Table 1: Average center location error (in pixels). The best two results are shown in red and blue fonts. The average fps follows in the end.

	Frag	TLD	IVT	DFT	ORIA	LOT	CXT	L1apg	MTT	ASLA	MIL	Struck	KCF	Ours
car4	131.5	12.8	2.9	61.9	37.4	67.3	58.1	16.4	**2.0**	4.3	50.8	8.7	19.1	**2.0**
girl	20.7	9.8	22.5	21.6	27.6	22.8	11.0	2.8	3.9	**2.7**	32.0	**2.6**	19.3	2.8
coke	124.8	25.1	83.0	**7.2**	50.0	69.4	25.7	50.4	30.0	60.2	21.0	12.1	13.6	**9.7**
deer	63.9	25.7	107.5	142.4	149.2	74.4	9.0	10.2	9.2	8.0	66.5	**7.5**	21.6	**4.5**
tiger1	74.3	**6.4**	106.6	6.5	87.0	31.4	45.4	58.4	64.4	55.9	15.0	12.8	69.9	**5.5**
couple	8.8	**2.5**	123.5	108.6	64.6	37.1	41.8	28.4	27.8	73.4	34.5	11.3	74.6	**7.2**
singer1	22.6	8.0	11.3	18.8	8.0	41.4	11.4	7.1	**1.4**	5.3	15.4	14.5	14.0	**5.0**
singer2	88.6	58.3	175.5	21.8	124.0	76.8	63.6	80.9	89.7	75.3	22.5	**15.3**	105.5	**12.5**
shaking	192.1	37.1	85.7	26.3	28.4	82.6	129.2	109.8	97.9	22.4	24.0	30.7	**17.2**	**5.6**
carScale	19.7	22.6	11.9	75.8	**7.9**	31.2	24.5	79.8	87.6	24.6	33.5	36.4	83.0	**7.5**
football	7.4	11.2	14.3	9.3	13.6	6.6	12.8	12.1	6.5	**6.1**	12.1	17.3	16.2	**4.8**
football1	15.7	45.4	24.5	**2.0**	63.8	6.8	2.6	9.2	12.7	12.2	5.6	5.4	16.5	**2.5**
walking2	57.5	44.6	**2.5**	29.1	20.0	64.9	34.7	5.1	4.0	37.4	35.6	11.2	17.9	**3.1**
sylvester	23.8	**5.9**	40.7	15.9	13.3	16.8	20.5	7.4	12.2	14.6	11.0	7.8	10.2	**5.6**
freeman1	**10.1**	39.7	11.6	10.4	96.1	86.9	26.8	62.4	117.8	25.7	11.2	14.3	125.5	**10.0**
freeman3	40.5	29.3	35.8	32.6	39.3	40.5	3.6	33.1	15.6	**3.2**	87.6	16.8	53.9	**3.0**
Speed	6 fps	28 fps	16 fps	13 fps	9 fps	1 fps	15 fps	2 fps	1 fps	2 fps	28 fps	15 fps	360fps	30 fps

Table 2: Average overlap rate(%). The best two results are shown in red and blue fonts.

	Frag	TLD	IVT	DFT	ORIA	LOT	CXT	L1apg	MTT	ASLA	MIL	Struck	KCF	Ours
car4	22	63	86	25	23	34	31	70	80	**89**	26	49	47	**90**
girl	45	57	17	28	24	42	55	73	66	71	40	**79**	57	**80**
coke	4	40	12	61	19	32	42	17	44	17	55	**67**	57	**72**
deer	17	60	22	26	24	20	70	60	61	73	21	74	**75**	**76**
tiger1	26	**70**	10	53	13	14	32	31	26	29	12	**70**	46	65
couple	57	**77**	7	8	4	45	48	47	49	28	50	54	28	**68**
singer1	34	73	66	35	65	19	49	28	34	**79**	36	36	36	**75**
singer2	20	22	4	**63**	13	26	17	24	34	34	51	24	24	**82**
shaking	8	39	13	**64**	44	13	12	28	34	46	43	35	58	**72**
carScale	42	45	63	41	66	35	**68**	50	49	61	41	41	41	**70**
football	70	56	56	66	51	66	54	68	**71**	57	59	53	55	**72**
football1	36	38	56	**87**	10	54	76	56	56	49	66	67	46	**78**
walking2	27	31	**80**	40	45	34	37	76	79	37	28	51	46	**82**
sylvester	**71**	67	52	60	65	57	63	55	65	59	53	58	60	**71**
freeman1	37	28	**43**	39	29	20	34	20	21	27	34	34	24	**47**
freeman3	32	44	39	31	36	12	71	35	46	**75**	21	26	30	**82**

are employed in our paper. A smaller average error or a bigger overlap rate means a more accurate result. Given the tracking result of each frame R_T and the corresponding ground truth R_G, we can get the overlap rate by the PASCAL VOC [14] criterion, $score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$. Table 1 and 2 report the quantitative comparison results respectively.

From Table 1 and 2, we can see clearly that our proposed method is comparable to other state-of-the-art trackers, especially when there is significant appearance changes and occlusion. Due to the limitation of pages, we compare our method with KCF and Struck on the remaining sequences of [43] in Table 3.

4.3. Qualitative evaluation

We also plot the results of Frag, TLD, IVT, DFT, ORIA, LOT, CXT, L1apg, MTT, ASLA, MIL, Struck and KCF trackers for qualitative comparison.

As an example of occlusion, the Walking2 sequence is captured in the shopping center with occlusion caused by a man. When the man appears in the video and occludes the target, KCF, MIL, CXT, TLD, ASLA and LOT start tracking the man instead of the target woman. L1apg, MTT, IVT and our method handle the problem well. Another example of occlusion is the Tiger1 sequence, all other trackers fail to track the tiger at frame 320 as it is partially occluded by the leaves. For the other sequences with occlusion such as Girl, Coke, Football and Football1, our method preforms reliably through the entire sequence. This can be attributed to the multiple part and adaptive updating schemes. When the target is partially occluded, the trackers of visible parts still work robustly. Once the occlusion disappears, because of the adaptive updating method, the classifiers of all the parts are able to relocate the target in the following frames.

In the sequences of Car4, CarScale, Singer1, Singer2, Freefman1 and Freeman3, the main challenge is the significant scale change. From Fig. 5(j), we can see that MTT, LOT, DFT, IVT and KCF failed to locate the car when it moves closer to the camera. However, benefiting from the multi-part and Bayesian framework, the proposed method adapts to the scale change and covers the car correctly till the end. Similar to the Car4 sequence, when the car drives far away from the camera, the scale of target decreases. Although the bounding box of KFC method still covers the

Table 3: Average center location error comparison with KCF and Struck.

	Basketball	Bolt	Boy	Cardark	Cross	David1	David2	David3	Dog	Doll	Dudek	Faceocc1	Faceocc2	Fish	Fleetface	Freeman4	Ironman
KCF	6.5	9.4	12.1	3.2	9	17.6	25.6	56.1	3.8	44.7	13.4	11.9	5.9	12.5	25.6	78.9	66.7
Struck	18.3	28.8	3.8	5.2	2.8	12.3	11.6	36.5	5.7	8.9	11.4	18.8	6	9	23	48.7	80.3
Our	6.3	8.2	3.5	3.1	7.6	8.5	10.3	16.4	4.2	8.5	10.9	10.8	5	8.7	8.5	20.1	73

	Jogging	Jumping	Lemming	Liquor	Matrix	Mhyang	Motor	MntBike	Skating1	Skiing	Soccer	Subway	Suv	Tiger2	Trellis	Walking	Woman
KCF	64.7	9.9	24.2	36.6	56.3	20.6	61.1	6.5	7.8	75.6	69.1	64.4	73.2	59.6	23.5	7.2	37.3
Struck	77.7	6.5	37.8	91	36.5	24.4	85.7	8.6	9	51.9	10.3	4.5	49.8	21.6	13.5	4.6	4.2
Our	14.6	7	22.4	15.6	26	17.3	67.8	6.5	7.9	86.2	55.6	4.3	9	12.3	10.6	5.3	4.1

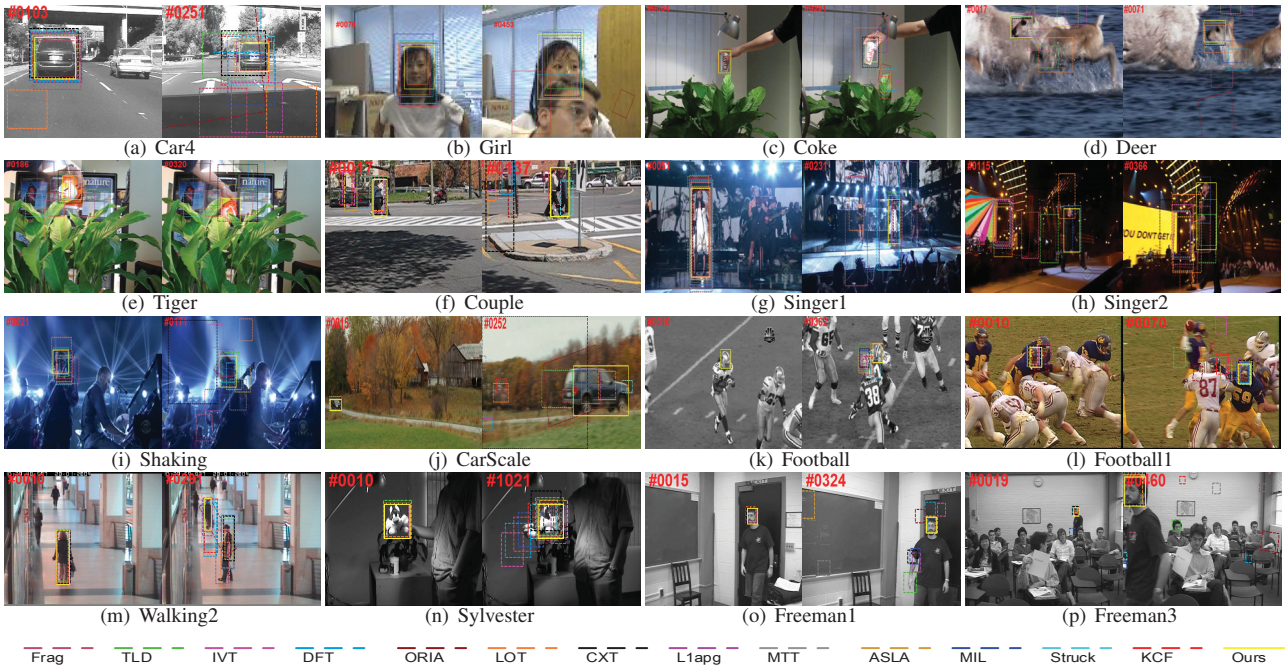


Figure 5: Comparison of our approach with state-of-the-art trackers in challenging situations.

target, because of the fixed scale size, the center location errors increase for the rest of the sequence. The proposed tracker handle the scale changes well.

In the Couple sequence, two people are tracked outdoor, which makes the appearance of target change significantly when the view is changed (as shown in Fig. 5(f)). The traditional KCF method failed at the beginning of the video. However, our part-based method treats the entire target as separate parts and succeeds to locate the target though parts. Frag and TLD also work well on this sequence. For the other sequences with illumination and fast movement, the proposed method also achieves better results and covers the target more correctly.

5. Conclusion

Based on the framework of correlation filter tracker and Bayesian inference, we developed a real-time part-based tracker with improved tracking performance. By using the adaptive weighting, updating and structural masking methods, our tracker is robust to occlusion, scale and appearance changes. Extensive experiments have been done to verify the reliability of our proposed method.

Acknowledgements: The research is supported by Singapore Ministry of Education (MOE) Tier 2 ARC28/14, and Singapore A*STAR Science and Engineering Research Council PSF1321202099. We also would like to thank Nvidia for their generous donation of GPU.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805. IEEE, 2006. 1, 6

[2] S. Avidan. Support vector tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1064–1072, 2004. 1

[3] S. Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*,

29(2):261–271, 2007. 1

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009. 1, 5, 6

[5] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1830–1837. IEEE, 2012. 6

[6] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic. Segmentation based particle filtering for real-time 2d object tracking. In *Computer Vision–ECCV 2012*, pages 842–855. Springer, 2012. 1

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010. 1, 2

[8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE, 2009. 2

[9] L. Cehovin, M. Kristan, and A. Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1363–1370. IEEE, 2011. 1

[10] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):941–953, 2013. 2, 6

[11] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1530–1537. IEEE, 2009. 2

[12] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*. IEEE, 2014. 2

[13] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1177–1184. IEEE, 2011. 6

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 7

[15] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10):1245–1256, 2013. 2, 6

[16] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011. 1, 5, 6

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015. 1, 2, 3, 6

[18] G. Hua and Y. Wu. Measurement integration under inconsistency for robust tracking. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 650–657. IEEE, 2006. 2, 6

[19] H. Izadinia, I. Saleemi, W. Li, and M. Shah. 2t: Multiple people multiple parts tracker. In *Computer Vision–ECCV 2012*, pages 100–114. Springer, 2012. 1

[20] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829. IEEE, 2012. 2, 6

[21] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56. IEEE, 2010. 1, 6

[22] S. Kwak, W. Nam, B. Han, and J. H. Han. Learning occlusion with likelihoods for visual tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1551–1558. IEEE, 2011. 1

[23] 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 *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1208–1215. IEEE, 2009. 2

[24] J. Kwon and K. M. Lee. Tracking by sampling trackers. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1195–1202. IEEE, 2011. 1

[25] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim. Visual tracking using pertinent patch selection and masking. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3486–3493. IEEE, 2014. 1

[26] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng. Visual tracking via incremental log-euclidean riemannian subspace learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[27] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Ro-

bust tracking using local sparse appearance model and k-selection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1313–1320. IEEE, 2011.

[28] T. Liu, G. Wang, L. Wang, and K. Chan. Visual tracking via temporally smooth sparse coding. *Signal Processing Letters, IEEE*, PP(99):1–1, 2014. 1

[29] S. Lucey. Enforcing non-positive weights for stable support vector tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[30] E. Maggio and A. Cavallaro. Multi-part target representation for color tracking. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–729. IEEE, 2005. 2

[31] X. Mei and H. Ling. Robust visual tracking using ℓ_1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443. IEEE, 2009. 2

[32] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1940–1947. IEEE, 2012. 6

[33] A. Rodriguez, V. N. Boddeti, B. V. Kumar, and A. Mahalanobis. Maximum margin correlation filter: A new approach for localization and classification. *Image Processing, IEEE Transactions on*, 22(2):631–643, 2013. 2

[34] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. 1, 2, 6

[35] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1910–1917. IEEE, 2012. 6

[36] S. Shahed Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2, 6

[37] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012. 1

[38] M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *ICCV*, 2011. 2

[39] B. Wang, G. Wang, K. L. Chan, and L. Wang. Tracklet association with online target-specific metric learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1234–1241, June 2014. 2

[40] D. Wang, H. Lu, and M.-H. Yang. Online object tracking with sparse prototypes. *Image Processing, IEEE Transactions on*, 22(1):314–325, 2013. 2

[41] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang. Video tracking using learned hierarchical features. *Image Processing, IEEE Transactions on*, 24(4):1424–1435, April 2015. 2

[42] X. Wang, G. Hua, and T. X. Han. Discriminative tracking by metric learning. In *Computer Vision–ECCV 2010*, pages 200–214. Springer, 2010. 1

[43] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE, 2013. 1, 7

[44] Y. Wu, B. Shen, and H. Ling. Online robust image alignment via iterative convex optimization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1808–1814. IEEE, 2012. 6

[45] W. Xiang and Y. Zhou. Part-based tracking with appearance learning and structural constrains. In *Neural Information Processing*, pages 594–601. Springer, 2014. 2

[46] J. Xing, J. Gao, B. Li, W. Hu, and S. Yan. Robust object tracking with online multi-lifespan dictionary learning. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 665–672. IEEE, 2013. 1

[47] B. Yang and R. Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Computer Vision–ECCV 2012*, pages 484–498. Springer, 2012. 1

[48] T. W. Yang Lu and S.-C. Zhu. Online object tracking, learning and parsing with and-or graphs. In *CVPR*, 2014. 2

[49] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Part-based visual tracking with online latent structural learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2363–2370. IEEE, 2013. 2, 6

[50] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006. 1

[51] K. Zhang and H. Song. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*, 46(1):397–411, 2013. 2

[52] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *Computer Vision–ECCV 2012*, pages 864–877. Springer, 2012. 1

[53] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International journal of computer vision*, 101(2):367–383, 2013. 6

[54] T. Zhang, K. Jia, C. Xu, Y. Ma, and N. Ahuja. Partial

occlusion handling for visual tracking via robust part matching. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1258–1265. IEEE, 2014. 2

[55] B. Zhong, X. Yuan, R. Ji, Y. Yan, Z. Cui, X. Hong, Y. Chen, T. Wang, D. Chen, and J. Yu. Structured partial least squares for simultaneous object tracking and segmentation. *Neurocomputing*, 133:317–327, 2014. 6

[56] Z. Zuo, G. Wang, B. Shuai, L. Zhao, and Q. Yang. Exemplar based deep discriminative and shareable feature learning for scene image classification. *Pattern Recognition*, 2015. 1