2014 22nd International Conference on Pattern Recognition

Visual tracking via saliency weighted sparse coding appearance model

Wanyi Li, Peng Wang
Research Center of Precision Sensing and Control
Institute of Automation, Chinese Academy of Sciences
Beijing, China
{Wanyi.li, peng_wang}@ia.ac.cn

Hong Qiao
State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences
Beijing, China

***Abstract*—Sparse coding has been used for target appearance modeling and applied successfully in visual tracking. However, noise may be inevitably introduced into the representation due to background clutter. To cope with this problem, we propose a saliency weighted sparse coding appearance model for visual tracking. Firstly, a spectral filtering based visual attention computational model, which combines both bottom-up and top-down visual attention, is proposed to calculate saliency map. Secondly, pooling operation in sparse coding is weighted by calculated saliency map to help target representation focus on distinctive features and suppress background clutter. Extensive experiments on a recently proposed tracking benchmark demonstrate that the proposed algorithm outperforms state-of-the-art methods in tracking objects under background clutter.**

***Keywords*—visual tracking; saliency; visual attention; sparse coding.**

I. INTRODUCTION

Visual tracking is an important computer vision task and is widely applied to robotics, surveillance, and human computer interaction, etc. Despite extensive research on this topic [1-3], robust tracking still remains a huge challenge due to different factors, such as occlusions, illumination changes and background clutter.

Recently, appearance modeling based on sparse coding (AMSC) [4-8] has been successfully applied in visual tracking and appealing experimental results are reported. The framework of AMSC contains three main layers [9]: image layer, coding layer and pooling layer. A set of local image patches are sampled in image layer. Each patch is sparsely coded by a learned dictionary in coding layer. To describe the appearance of the input image, produced codes in coding layer are further pooled in the pooling layer and form the final feature vector.

Real-world video frames almost always contain background clutter. As pooling operation selects features without notion of foreground, features extracted from background may have stronger response to the learned dictionary. As a result, the pooling operation alone may inevitably introduce noise into the representation. Thus corresponding tracking algorithms may fail when background clutter occurs.

Human visual system (HVS) has robust visual tracking capability. In tracking process of HVS, visual attention plays a critical role, which directs processing resources to potentially most relevant visual data such as foreground regions, especially directs our gaze rapidly towards objects of interest. As a result, humans can easily achieve robust tracking. In the computer vision community, many computational models have been proposed to simulate human's visual attention [10]. The output of these visual attention computational models is saliency map of which saliency is a good indicator of foreground. As a result, if we use visual saliency to guide pooling operation in sparse coding, better sparse representation focusing on foreground regions can be obtained.

Motivated by above-mentioned discussions, this paper proposes a saliency weighted sparse coding appearance model for tracking objects under background clutter. The contributions of this work are summarized as follows. First, a spectral filtering based visual attention computational model, which combines both top-down and bottom-up visual attention, is proposed to calculate saliency map. Second, we propose a saliency weighted pooling function and lead to a saliency weighted sparse coding appearance model for visual tracking. Extensive experiments on challenging sequences demonstrate the effectiveness of the proposed method.

II. PROPOSED METHOD

A. Spectral Filtering based Visual Attention Computational Model

(a). Spectral Filtering

The convolution of the image amplitude spectrum with a low-pass Gaussian kernel of an appropriate scale is equivalent to an image saliency detector [11]. The key ideas include two parts: 1) Spikes in the amplitude spectrum correspond to repeated patterns. 2) Repeated patterns can be suppressed by spectral filtering, thus the saliency pops out from the rest of the image. The saliency map can be obtained by reconstructing the 2-D signal using the original phase and the filtered amplitude spectrum, shown as Eq. (1)-(2).

$$\mathrm{A}_S(u,v) = \left|\mathrm{F}\left\{\mathrm{f}(x,y)\right\}\right| * \mathrm{g}, \tag{1}$$

$$S = \mathrm{F}^{-1}\left\{\mathrm{A}_S(u,v) e^{i\cdot \mathrm{P}(u,v)}\right\}, \tag{2}$$

This work was supported by the NNSF (National Natural Science Foundation) of China under the grant 61100098, 61379097, 61033011 and 61210009 .

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.701

where $\mathrm{F}\{\mathrm{f}\}$ and $\mathrm{F}^{-1}\{\mathrm{f}\}$ indicates the Fourier transform and inverse Fourier transform of the image $\mathrm{f}(x,y)$ respectively, g is a low-pass Gaussian kernel, $\mathrm{f}*\mathrm{g}$ indicates the convolution of f and g, $|x|$ represents the amplitude of complex number x, $\mathrm{P(u,v)}$ is the phase of Fourier transform of image $\mathrm{f}(x,y)$.

(b). Extraction of Early Visual Features

Given an input image, let r, g, and b denote the red, green, and blue channels respectively, then the intensity image I can be computed as $\mathrm{I}=(\mathrm{r}+\mathrm{g}+\mathrm{b})/3$, and the inverse intensity image can be computed as $\mathrm{I}_{off}=255-\mathrm{I}$. Four broadly-tuned color channels are created for red, green, blue and yellow respectively, i.e., $\mathrm{R}=\mathrm{r}-(\mathrm{g}+\mathrm{b})/2$, $\mathrm{G}=\mathrm{g}-(\mathrm{r}+\mathrm{b})/2$, $\mathrm{B}=\mathrm{b}-(\mathrm{r}+\mathrm{g})/2$, and $\mathrm{Y}=(\mathrm{r}+\mathrm{g})/2-|\mathrm{r}-\mathrm{g}|/2-\mathrm{b}$ in which negative values are set to zero. Accordingly, map RG is created to simultaneously account for red/green and green/red double opponency and BY for blue/yellow and yellow/blue double opponency, which are calculated by Eq. (3) and Eq. (4).

$$\mathrm{RG}=|\mathrm{R}-\mathrm{G}|, \tag{3}$$

$$\mathrm{BY}=|\mathrm{B}-\mathrm{Y}|. \tag{4}$$

(c). Bottom-up Saliency Map

After applying spectral filtering to feature channels RG, BY and I, resulted feature maps $\mathrm{F}_j, j\in\{\mathrm{RG},\mathrm{BY},\mathrm{I}\}$ are linearly summed and normalized to yield the bottom-up saliency map, shown as Eq. (5).

$$\mathrm{S}_{bu}=\frac{1}{3}\sum_j \mathrm{F}_j,\ j\in\{\mathrm{RG},\mathrm{BY},\mathrm{I}\}. \tag{5}$$

(d). Top-down Saliency Map

The top-down saliency map is calculated by Eq. (6)-(8). When new frame $\mathrm{f}(t)$ comes, the previously learned weight vector $\mathbf{w}$ (see the paragraph at the end of this section for detailed learning procedure) is used to weight the feature maps $\{\mathrm{F}_j\}$, which are determined by spectral filtering. Depending on the values w_j, the maps are used to compute the excitation map E or the inhibition map I. E is the weighted sum of all feature maps S_j that are important for the learned region, i.e., $w_j > 1$:

$$\mathrm{E}(\mathrm{f}(t))=\sum_j w_j\,\mathrm{F}_j(\mathrm{f}(t)), w_j>1, \forall j\in\{R,G,B,Y,I,I_{off}\} \tag{6}$$

The inhibition map I considers the features more present in the background than in the target region, i.e., $w_j<1$:

$$\mathrm{I}(\mathrm{f}(t))=\sum_j (\frac{1}{w_j})\times \mathrm{F}_j(\mathrm{f}(t)), w_j<1, \forall j\in\{R,G,B,Y,I,I_{off}\} \tag{7}$$

Maps with $w_j=1$ are completely unimportant for the target and are ignored. The top-down saliency map S_{td} results from the difference of E and I and a clipping of negative values:

$$\mathrm{S}_{td}(\mathrm{f}(t))=\max(\mathrm{E}(\mathrm{f}(t))-\mathrm{I}(\mathrm{f}(t)),\mathbf{0}). \tag{8}$$

The weight vector $\mathbf{w}=(w_R,w_G,w_B,w_Y,w_I,w_{I_{off}})^T$ representing salient features of the target object relate to its surrounding is computed at the first frame $\mathrm{f}(0)$ as Eq. (9). The value w_j for feature map F_j is computed as the ratio between the mean saliency of target region and background:

$$w_j=\frac{\mathrm{mean}(\mathrm{F}_j(\mathrm{T}(0)))}{\mathrm{mean}(\mathrm{F}_j(\mathrm{f}(0)\setminus\mathrm{T}(0)))}, j\in\{R,G,B,Y,I,I_{off}\}, \tag{9}$$

where $\mathrm{T}(0)$ denotes target region and $\mathrm{f}(0)\setminus\mathrm{T}(0)$ denotes background region.

(e). Final Saliency Map Combined Bottom-up and Top-down Visual Attention

The final saliency map is computed as the linear combination of the bottom-up saliency map and the top-down saliency map, shown as Eq. (10).

$$\mathrm{S}=\alpha\,\mathrm{S}_{td}+(1-\alpha)\mathrm{S}_{bu}, \alpha\in[0,1], \tag{10}$$

where α is the top-down coefficient, which is used to tune the relative importance of top-down visual attention and bottom-up visual attention.

B. Saliency Weighted Sparse Coding Apperance Model

(a). Saliency Pooling

Pooling operation is an essential step in the general framework of appearance modeling based on sparse coding (AMSC) for visual tracking. As traditional pooling operation selects features without notion of foreground, features extracted from background may have stronger response to the learned dictionary. As a result, the pooling operation alone may inevitably introduce noise into the representation. To cope with the problem, we propose a saliency weighted pooling function to help pooling operation focus on regions where foreground may appear. The traditional pooling operation is shown in Eq. (11) and the saliency weighted pooling function is shown in Eq. (12). These two pooling operation result in final pooled feature vectors $\mathbf{f}$ and $\mathbf{f}_s$.

$$\mathbf{f}=\mathrm{pooling}(\mathbf{B}), \tag{11}$$

$$\mathbf{f}_s=\mathrm{swPooling}(\mathbf{B})=\mathrm{pooling}(\mathbf{B}_s), \tag{12}$$

where $\mathbf{B}=[\mathbf{b}_1,\mathbf{b}_2,\cdots,\mathbf{b}_N]$ represents the sparse codes of one candidate. $\mathbf{b}_i, i\in\{1,2,\cdots,\mathrm{N}\}$ is the sparse code of the i^{th} local image patch in the candidate. $\mathbf{B}_s=[\mathrm{s}_1\,\mathbf{b}_1,\mathrm{s}_2\,\mathbf{b}_2,\cdots,\mathrm{s}_N\,\mathbf{b}_N]$ is the saliency weighted sparse codes. $s_i=\mathrm{mean}(\mathrm{S}(\mathrm{y}_i)), i\in\{1,2,\cdots,\mathrm{N}\}$ is the mean saliency of the i^{th} local image patch y_i.

(b). Saliency Weighted Sparse Coding

To demonstrate the advantage of the above proposed saliency pooling function, we introduce the proposed saliency pooling function to the adaptive structural local sparse appearance model [8] (ASLA) as a case study.

Given target templates $\mathbf{T} = [\mathbf{T}_1, \ldots, \mathbf{T}_n]$, a set of overlapped local image patches are sampled inside the target region. To encode the local patches inside the possible candidate regions, image patches sampled from target templates are used as the dictionary, i.e., $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2 \ldots, \mathbf{d}_{(n \times N)}] \in \mathrm{R}^{d \times (n \times N)}$, where n is the number of target templates, N is the number of local patches sampled from the target region and d is the dimension of the image patch vector. By using l_2 normalization on the vectorized image patches sampled from $\mathbf{T}$, each column of $\mathbf{D}$ is obtained. For a target candidate, N local image patches are extracted in the same way, and denoted as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathrm{R}^{d \times N}$.

Under a sparse constraint, sparse coding represents the i^{th} local patch $\mathbf{y}_i$ within the target region as a linear combination of only a few basis elements of dictionary $\mathbf{D}$. The sparse code $\mathbf{b}_i \in \mathrm{R}^{(n \times N) \times 1}$ of $\mathbf{y}_i$ can be solved by

$$\mathbf{b}_i = \arg\min_{\mathbf{b}} \|\mathbf{y}_i - \mathbf{D}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1 \quad s.t.\ \mathbf{b} \succeq \mathbf{0}. \qquad (13)$$

The sparse codes of one candidate are denoted as $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_N]$.

The sparse codes are then pooled by the alignment pooling operator, which computes the feature element for each patch as the sum of the codes corresponding to the basis elements at the same position with the patch and is defined as follows. The sparse code $\mathbf{b}_i \in \mathrm{R}^{(n \times N) \times 1}$ of $\mathbf{y}_i$ is divided into n segments, i.e., $\mathbf{b}_i^T = [\mathbf{b}_i^{(1)T}, \mathbf{b}_i^{(2)T}, \ldots, \mathbf{b}_i^{(n)T}]$, where $\mathbf{b}_i^{(k)T} \in \mathrm{R}^{N \times 1}, k \in \{1, 2, \cdots, \mathrm{n}\}$ is the k^{th} segment of the coefficient vector $\mathbf{b}_i$ corresponding to template $\mathbf{T}_k$. These segmented coefficients are linear summed to obtain $\mathbf{v}_i$ for the i^{th} patch,

$$\mathbf{v}_i = \frac{1}{C} \sum_{k=1}^{n} \mathbf{b}_i^{(k)}, i = 1, 2, \ldots, N, \qquad (14)$$

where $\mathbf{v}_i$ corresponds to the i^{th} local patch and C is a normalization constant. All the vectors $\mathbf{v}_i$ of local patches in a candidate region form a square matrix $\mathbf{V}$, and the final pooled feature is obtained by taking the diagonal elements of the square matrix $\mathbf{V}$,

$$\mathbf{f} = \mathrm{pooling}(\mathbf{B}) = \mathrm{Diag}(\mathbf{V}). \qquad (15)$$

Using the proposed saliency weighted pooling function, i.e., Eq. (12) to Eq. (15), the pooled feature with saliency property is calculated as Eq. (16).

$$\mathbf{f}_s = \mathrm{swPooling}(\mathbf{B}) = \mathrm{pooling}(\mathbf{B}_s) = \mathrm{Diag}(\mathbf{V}_s), \qquad (16)$$

where $\mathbf{B}_s = [\mathrm{s}_1 \mathbf{b}_1, \mathrm{s}_2 \mathbf{b}_2, \cdots, \mathrm{s}_N \mathbf{b}_N]$, $\mathbf{V}_s = [s_1 \mathbf{v}_1, s_2 \mathbf{v}_2, \ldots, s_N \mathbf{v}_N]$, $s_i = \mathrm{mean}(\mathrm{S}(\mathbf{y}_i)), i \in \{1, 2, \cdots, \mathrm{N}\}$ is the mean saliency of the i^{th} local image patch y_i .

C. Tracking Algrithm

The proposed appearance model is embedded within the Bayesian tracking framework. Based on the set of all available measurements $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$, the target state $\mathbf{x}_t$ can be estimated as

$$\hat{\mathbf{x}}_t = \arg\max_{x_t^i} p(\mathbf{x}_t^i \mid \mathbf{z}_{1:t}), \qquad (17)$$

where $\mathbf{x}_t^i$ is the state of the i^{th} sample. With the Bayesian theorem, the posterior probability $p(\mathbf{x}_t \mid \mathbf{z}_{1:t})$ can be inferred recursively as Eq. (18).

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t}) \propto p(\mathbf{z}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}, \qquad (18)$$

where $p(\mathbf{z}_t \mid \mathbf{x}_t)$ indicates the observation model and $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ represents the dynamic model. The target motion between two consecutive frames is modeled by affine transformation with six parameters. The state transition is formulated as $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = N(\mathbf{x}_t; \mathbf{x}_{t-1}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a diagonal covariance matrix. The observation model is constructed by Eq. (19).

$$p(\mathbf{z}_t \mid \mathbf{x}_t) \propto \sum_{k=1}^{N} \mathbf{f}_s^{(\mathrm{k})}, \qquad (19)$$

where $\sum_{k=1}^{N} \mathbf{f}_s^{(\mathrm{k})}$ denotes the similarity between the candidate and the target based on the pooled feature vector $\mathbf{f}_s$ calculated using Eq. (16), and $\mathbf{f}_s^{(\mathrm{k})}$ is the k^{th} component of the pooled feature vector $\mathbf{f}_s$.

For template update, we use the update scheme of ASLA [8], which exploit both sparsity and subspace learning.

III. Experiments

Our tracker is implemented in MATLAB which runs at 1.5 fps on a Intel(R) 2.93 GHz Dual Core PC with 2GB memory. For each sequence, the location of the target object is initialized as the ground truth position in the first frame. The top-down coefficient α in Eq. (10) is set to 0.5. The other parameters are same as the default of ASLA [8]. Since the proposed saliency pooling function is applied to adaptive structural local sparse appearance model [8] (ASLA) as a case study, our method is referred to as SWASLA (means Saliency Weighted ASLA).

To evaluate our approach (SWASLA), we compare it with other methods on the tracking benchmark proposed just recently in [1] including 50 videos and 29 methods. For qualitative comparison, we compare the proposed method with the baseline method, i.e., ASLA. For quantitative evaluation, we compare our method with 29 methods of the benchmark.

A. *Qualitative Comparison*

Fig.1 – Fig.4 show the comparative tracking results of ASLA and SWASLA on four videos with serious background clutter. Fig.1 shows the tracking results of "*Basketball*" sequence. ASLA drifts away from the target since the 460th frame. Fig.2 shows the tracking results of "*Singer2*" sequence, from the 20th frame, ASLA loses the target. Fig.3 shows the tracking results of "*Soccer*" sequence, ALSA is distracted by a similar image region at the 66th frame. Fig.4 shows the tracking results of "*Subway*" sequence, ALSA drifts at the 42th frame and distracted by another man. In these sequences, the proposed method SWASLA can always lock onto the object while ASLA loses target to some extent.

B. *Quantitative Evaluation*

The success plot metric [1] are used for a quantitative evaluation. The success plot shows the ratios of successful frames at the thresholds of overlap score varied from 0 to 1. Given tracked bounding box S_t and ground truth bounding box S_t^{GT}, the overlap score is defined as $SC_t = \left|S_t \cap S_t^{GT}\right| / \left|S_t \cup S_t^{GT}\right|$, where ∩ and ∪ represent the intersection and union of two regions respectively, and $|.|$ denotes the number of pixels in the region. A frame with overlap score larger than a given threshold will be counted as a successful tracked frame. Area under curve (AUC) scores are used to summarize and rank the trackers.

For robustness evaluation, we run trackers in two ways, one-pass evaluation (OPE) and spatial robustness evaluation (SRE). OPE is the conventional way to evaluate trackers, which is to run trackers throughout a test sequence with initialization from the ground truth position in the first frame and report the success rate. SRE analyze a tracker's robustness to initialization by perturbing the initialization spatially (i.e., start by different bounding boxes).

Fig.5 summarizes the overall success plots of OPE and SRE. For SWASLA, AUC of OPE on the left is the third while AUC of SRE on the right is the second of all methods, outperforming the ASLA. Fig.6 shows the success plots of OPE and SRE on background clutter subset, SWASLA outperforms the other methods.

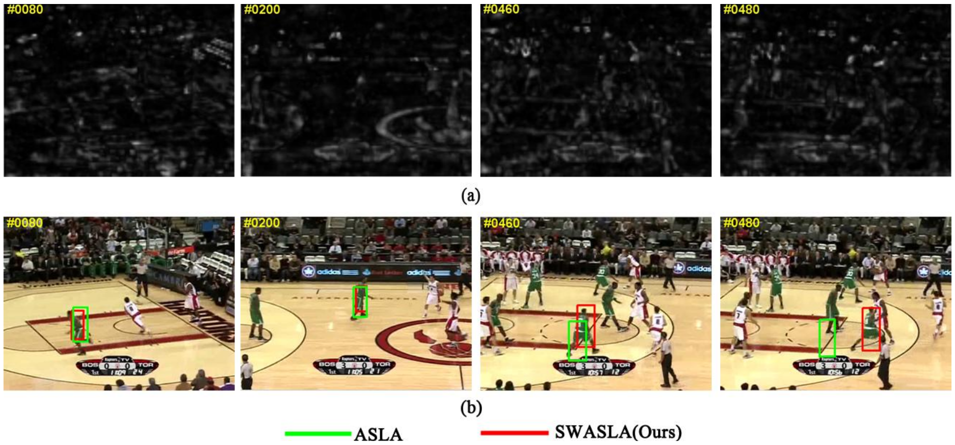

Fig.1. Comparative tracking results on "*Basketball*" sequence. Frame No: 80, 200, 460, 480. (a) Saliency map calculated by proposed visual attention model. (b) Tracking results. Red rectangle represents for the tracking result of our method, SWASLA, while green rectangle denotes ASLA's.

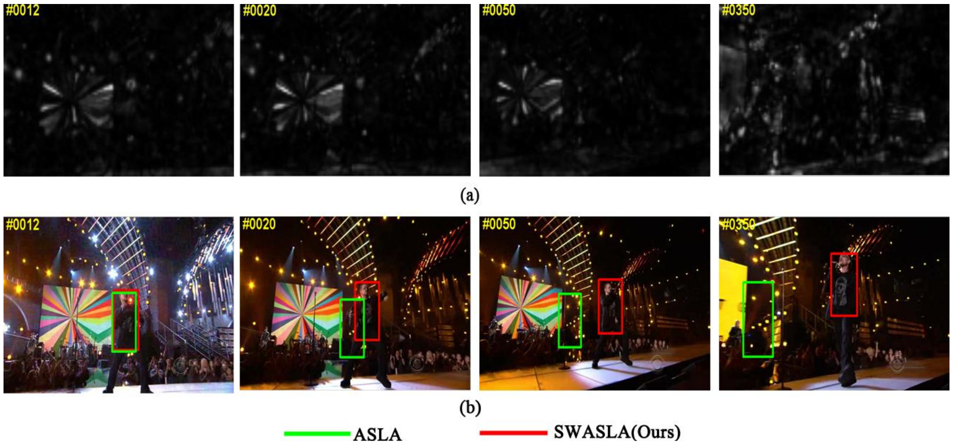

Fig.2. Comparative tracking results on "*Singer2*" sequence. Frame No: 12, 20, 50, 350. (a) Saliency map calculated by proposed visual attention model. (b) Tracking results. Red rectangle represents for the tracking result of our method, SWASLA, while green rectangle denotes ASLA's.

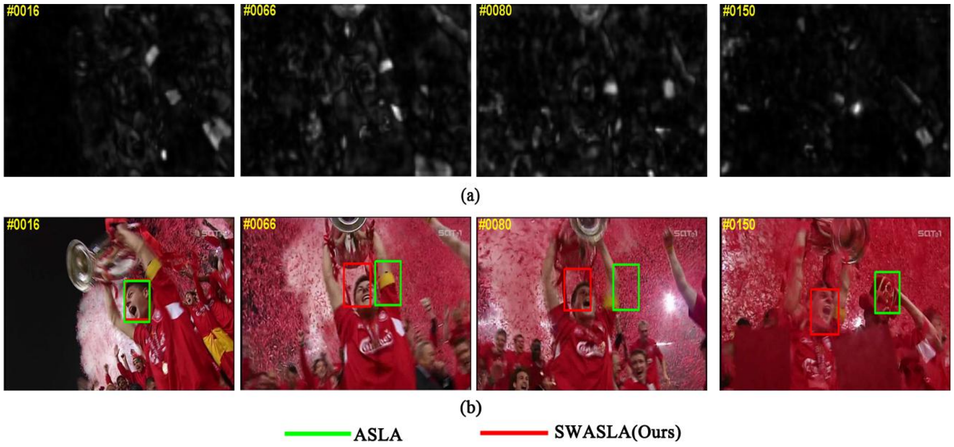

Fig.3. Comparative tracking results on "*Soccer*" sequence. Frame No: 16, 66, 80, 150. (a) Saliency map calculated by proposed visual attention model. (b) Tracking results. Red rectangle represents for the tracking result of our method, SWASLA, while green rectangle denotes ASLA's.

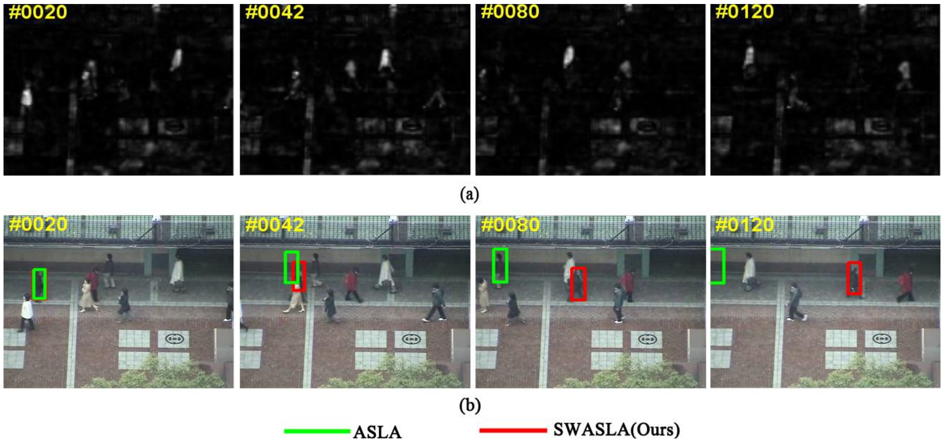

Fig.4. Comparative tracking results on "*Subway*" sequence. Frame No: 20, 42, 80, 120. (a) Saliency map calculated by proposed visual attention model. (b) Tracking results. Red rectangle represents for the tracking result of our method, SWASLA, while green rectangle denotes ASLA's.

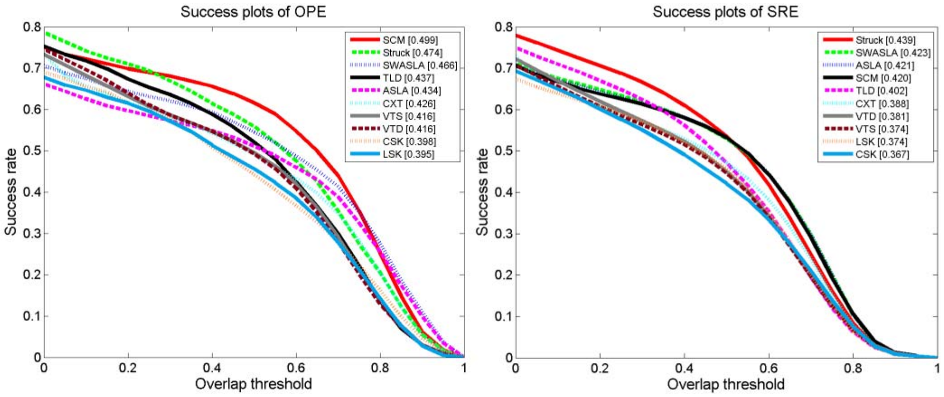

Fig.5. Success plots of OPE and SRE. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity (best viewed on high-resolution display)

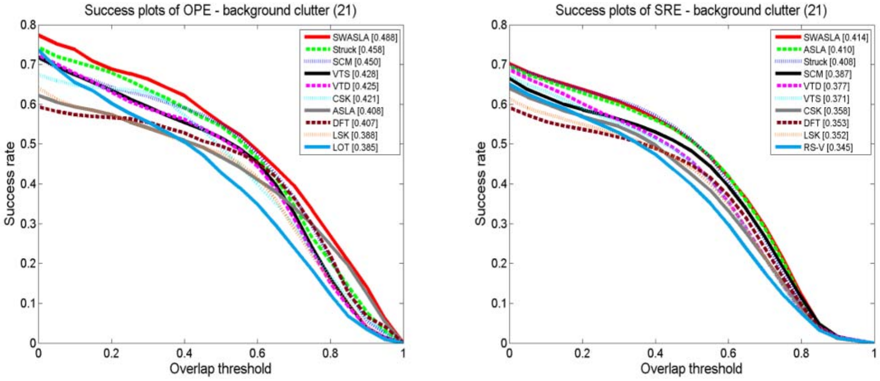

Fig.6. Success plots of OPE and SRE of background clutter subset. The value appears in the title is the number of sequences in this subset, i.e., there are 21 sequences in this subset. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity (best viewed on high-resolution display)

IV. Conclusion

This paper proposes a saliency weighted sparse coding appearance model for visual tracking. A novel spectral filtering based visual attention computational model calculates saliency map and calculated saliency map is used to weight the pooling operation in sparse coding. Experimental results on a recently proposed tracking benchmark show the effectiveness of the proposed method. Firstly, the presented method outperforms baseline tracker in overall performance. Secondly, the proposed method can cope with background clutter robustly and outperforms state-of-the-art methods on background clutter subset.

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online Object Tracking: A Benchmark," presented at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.,* vol. 38, p. 13, 2006.

[3] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing,* vol. 74, pp. 3823-3831, 2011.

[4] S. P. Zhang, H. X. Yao, and S. H. Liu, "Robust visual tracking using feature-based visual attention," in *2010 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ed New York: IEEE, 2010, pp. 1150-1153.

[5] Q. Wang, F. Chen, J. Yang, W. Xu, and M.-H. Yang, "Transferring visual prior for online object tracking," *Image Processing, IEEE Transactions on,* vol. 21, pp. 3296-3305, 2012.

[6] B. Liu, J. Huang, L. Yang, and K. C., "Robust tracking using local sparse appearance model and K-selection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 1313-1320.

[7] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, 2012, pp. 425-432.

[8] J. Xu, L. Huchuan, and Y. Ming-Hsuan, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 1822-1829.

[9] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognition,* vol. 46, pp. 1772-1788, 2013.

[10] A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 35, pp. 185-207, 2013.

[11] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual Saliency Based on Scale-Space Analysis in the Frequency Domain," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 35, pp. 996-1010, 2013.