Visual Tracking via Adaptive Structural Local Sparse Appearance Model

Xu Jia
Dalian University of Technology
jiaxu1986@mail.dlut.edu.cn

Huchuan Lu
Dalian University of Technology
lhchuan@dlut.edu.cn

Ming-Hsuan Yang
University of California at Merced
mhyang@ucmerced.edu

Abstract

Sparse representation has been applied to visual tracking by finding the best candidate with minimal reconstruction error using target templates. However most sparse representation based trackers only consider the holistic representation and do not make full use of the sparse coefficients to discriminate between the target and the background, and hence may fail with more possibility when there is similar object or occlusion in the scene. In this paper we develop a simple yet robust tracking method based on the structural local sparse appearance model. This representation exploits both partial information and spatial information of the target based on a novel alignment-pooling method. The similarity obtained by pooling across the local patches helps not only locate the target more accurately but also handle occlusion. In addition, we employ a template update strategy which combines incremental subspace learning and sparse representation. This strategy adapts the template to the appearance change of the target with less possibility of drifting and reduces the influence of the occluded target template as well. Both qualitative and quantitative evaluations on challenging benchmark image sequences demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.

1. Introduction

Visual tracking has long been an important topic in computer vision field, especially for application of surveillance, vehicle navigation and human computer interface. Although many tracking methods have been proposed, it remains a challenging problem due to factors such as partial occlusions, illumination changes, pose changes, background clutter and viewpoint variation.

Current tracking algorithms can be categorized into either generative or discriminative approaches. Discriminative methods formulate tracking as a classification problem which aims to distinguish the target from the background. It employs the information from both the target and background. Avidan [2] combines a set of weak classifiers into a strong one to do ensemble tracking. In [7] Grabner *et al*. propose an online boosting method to update discriminative features and later in [8] a semi-online boosting algorithm is proposed to handle the drifting problem. Babenko *et al*. [3] use multiple instance learning (MIL) which puts all ambiguous positive and negative samples into bags to learn a discriminative model for tracking. Kalal *et al*. [9] propose the P-N learning algorithm to exploit the underlying structure of positive and negative samples to learn effective classifiers for object tracking. Wang *et al*. [20] base the discriminative appearance model on superpixels, which facilitates the tracker to distinguish between the target and background.

Generative methods formulate the tracking problem as searching for the regions most similar to the target model. These methods are based on either templates [13, 5, 15, 1, 10] or subspace models [4, 18]. To adapt to the target appearance variations caused by pose change and illumination change, the target appearance model is updated dynamically. Matthews *et al*. [15] develop a template update method which can reduce the drifting problem by aligning with the first template to reduce drifts. In [18], the low-dimensional subspace representation is learned incrementally during the tracking process to adapt to the changes of target appearance. Kwon *et al*. [10] decompose the observation model into multiple basic observation models to cover a wide range of pose and illumination variation. Most of these methods use the holistic model to represent the target and hence cannot handle partial occlusion or distracters.

Recently, several tracking methods based on sparse representation have been proposed [16, 12, 17, 11]. Mei *et al*. [16, 17] adopt the holistic representation of the object as the appearance model and then track the object by solving the ℓ_1 minimization problem. Liu *et al*. [11] propose a tracking algorithm based on local sparse model which employs histograms of sparse coefficients and the mean-shift algorithm for object tracking. However, this method is based on a static local sparse dictionary and may fail when there is similar object in the scenes.

In this paper, we propose an efficient tracking algorithm with structural local sparse model and adaptive template up-

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

date strategy. The proposed method samples overlapped local image patches within the target region. We observe that sparse coding of local image patches with a spatial layout contains both spatial and partial information of the target object. The similarity measure is obtained by proposed alignment-pooling method across the local patches within one candidate region. This helps locate the target more accurately and handle partial occlusion. In addition, the dictionary for local sparse coding is generated from the dynamic templates, which are updated online based on both incremental subspace learning and sparse representation. The update scheme facilitates the tracker to account for appearance changes of the target. Due to the simplicity of appearance model and template update strategy, our method can track the target efficiently.

The contributions of this work are summarized as follows. First, sparse codes of local image patches with spatial layout in an object are used to model its appearance model. As for the sparse codes, we propose an alignment-pooling method to improve accuracy of tracking and reduce the influence of occlusion as well. Second, both incremental subspace learning and sparse representation are employed to update the templates to handle the drifting problem and partial occlusion. Experiments on challenging benchmark image sequences demonstrate that the proposed tracking approach performs favorably against several state-of-the-art methods.

2. Related Work and Context

Sparse representation has been successfully applied in numerous vision applications [21, 16, 12, 17, 11]. With sparsity constraints, one signal can be represented in the form of linear combination of only a few basis vectors. In [16, 17], the target candidate is sparsely represented as a linear combination of the atoms of a dictionary which is composed of dynamic target templates and trivial templates. By introducing trivial templates, the tracker can handle partial occlusion. This sparse representation problem is then solved through ℓ_1 minimization with non-negativity constraints. In [12], dynamic group sparsity which includes both spatial and temporal adjacency is introduced into the sparse representation to enhance the robustness of the tracker. In [11], a local sparse representation scheme is employed to model the target appearance and then represent the basis distribution of the target with the sparse coding histogram. Due to the representation of local patches, their method performs well especially in handling the partial occlusion. However, histograms of local sparse coefficients alone cannot provide enough spatial information. A mean-shift algorithm [5] and a sparse-representation-based voting map are used to better track the target.

Our work bears some similarity to [11] in the use of local sparse representations. However, we sample larger overlapped local image patches with fixed spatial layout where there are more spatial structural information in them. In addition, we make full use of the sparse coding coefficients with the proposed alignment-pooling method rather than histograms and kernel densities to measure the similarity. Instead of using fixed template [1] or dictionary [11] learned from the first frame, we update the dictionary adaptively using dynamic templates. Object tracking with a static template is likely to fail in dynamic scenes due to large appearance change. In [16, 17], the template is updated according to both the weights assigned to each template and the similarity between templates and current estimation of target candidate. Different from that template update scheme, we employ both incremental subspace learning and sparse representation to update the templates adaptively. This template update method reduces the drifting problem and puts more weights on the important parts of the target. In addition, it reduces the influence of the template with partial occlusion.

3. Structural Local Sparse Appearance Model

Given the image set of the target templates $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_n]$, we sample a set of overlapped local image patches inside the target region with a spatial layout. These local patches are used as the dictionary to encode the local patches inside the possible candidate regions, i.e. $\mathbf{D} = \left[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{(n\times N)}\right] \in \mathbb{R}^{d\times(n\times N)}$, where d is the dimension of the image patch vector, n is the number of target templates and N is the number of local patches sampled within the target region. Each column in $\mathbf{D}$ is obtained by ℓ_2 normalization on the vectorized local image patches extracted from $\mathbf{T}$. Each local patch represents one fixed part of the target object, hence the local patches altogether can represent the complete structure of the target. Since the local patches are collected from many templates, this dictionary captures the commonality of different templates and is able to represent various forms of these parts. For a target candidate, we extract local patches within it and turn them into vectors in the same way, which are denoted by $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathbb{R}^{d\times N}$.

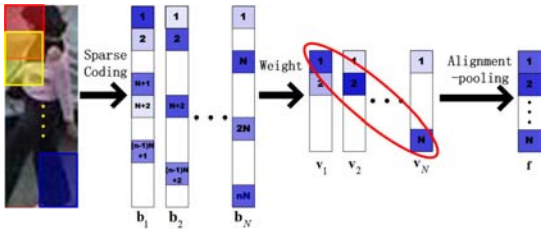


Figure 1. Illustration of feature formation by alignment-pooling (darker color elements have larger values).

With the sparsity assumption, the local patches within the target region can be represented as the linear combina-

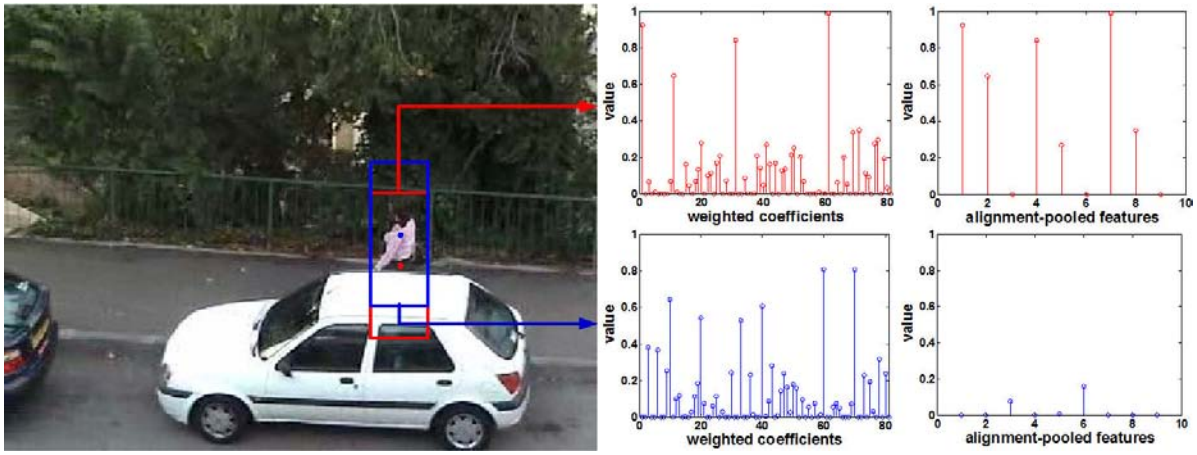


Figure 2. Comparison of the pooled features obtained by alignment-pooling as for good and bad candidates. The upper and lower rows show the pooled features for a good candidate (i.e., a region close to ground-truth tracking result) and a bad candidate (i.e., a region with large tracking error).

tion of only a few basis elements of the dictionary by solving

$$\min_{\mathbf{b}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{b}_i\|_2^2 + \lambda\|\mathbf{b}_i\|_1, \quad \text{s.t. } \mathbf{b}_i \succcurlyeq 0 \tag{1}$$

where $\mathbf{y}_i$ denotes the i-th vectorized local image patch, $\mathbf{b}_i \in \mathbb{R}^{(n\times N)\times 1}$ is the corresponding sparse code of that local patch, and $\mathbf{b}_i \succcurlyeq 0$ means all the elements of $\mathbf{b}_i$ are non-negative. Note $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$ represents the sparse codes of one candidate. The sparse coefficients of each local patch are divided into several segments, according to the template that each element of the vector corresponds to, i.e., $\mathbf{b}_i^\top = \left[\mathbf{b}_i^{(1)\top}, \mathbf{b}_i^{(2)\top}, \ldots, \mathbf{b}_i^{(n)\top}\right]$, where $\mathbf{b}_i^{(k)} \in \mathbb{R}^{N\times 1}$ denotes the k-th segment of the coefficient vector $\mathbf{b}_i$. These segmented coefficients are weighted to obtain $\mathbf{v}_i$ for the i-th patch,

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

where vector $\mathbf{v}_i$ corresponds to the i-th local patch and C is a normalization term. As the templates contain the target object with some appearance variation, the blocks that appear frequently in these templates (as indicated by their sparse codes) should be weighted more than others for more robust representation. This weighting process is carried out by Eq. 2 with their sparse codes. All the vectors $\mathbf{v}_i$ of local patches in a candidate region form a square matrix $\mathbf{V}$ and further processed with a novel pooling method.

Although for a single local patch we lose spatial information by considering only its own coefficient vector as described above, we alleviate this problem by using a novel method to pool the responses of local patches within the candidate region. We propose an alignment-pooling algorithm rather than max-pooling method [22] to improve the accuracy of location estimation. After obtaining $\mathbf{v}_i$, each local patch at a certain position of the candidate is represented by patches at different positions of the templates. The local appearance variation of a patch can be best described by the blocks at the same positions of the template (i.e., using the sparse codes with the aligned positions). For example, the top left corner patch of the target object in Figure 1 should be best described by the first element of $\mathbf{v}_1$ as it should have the largest coefficient value (via Eq 2 and its block location). Therefore, we take the diagonal elements of the square matrix $\mathbf{V}$ as the pooled feature, i.e.,

$$\mathbf{f} = \text{diag}(\mathbf{V}), \tag{3}$$

where $\mathbf{f}$ is the vector of pooled features. Since the weighting operation increase the stability of sparse coding, this pooling method further aligns local patterns between target candidate and the templates based on the locations of structural blocks. The aligned tracking results also facilitate the incremental subspace learning for template update in our algorithm. The proposed representation with alignment-pooling process captures structural information of a target object in terms of blocks. In addition, this appearance model is able deal with partial occlusion. When occlusion occurs, the appearance change makes the representation of the occluded local patches dense. However, the local patches which are not occluded still have sparse representations.

After pooling across these local patches, the influence of outliers is reduced and the structural information is retained in the representation to better locate the target. Figure 2 shows the vector $\mathbf{v}_i$ and pooled features obtained by our method for good and bad target candidates. When the target object is partial occlusion, the image patches which are not occluded can still be represented by only few atoms of the dictionary with large coefficients whereas the occluded

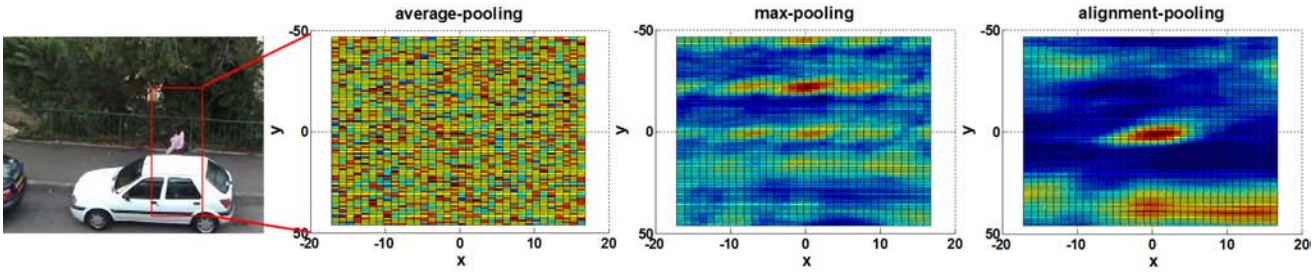


Figure 3. Comparison of the confidence map obtained by three kinds of pooling methods within a range around the target object. Red color blocks denote large coefficient (confidence) values and blue color ones denote low values. The resulting confidence map using our representation indicates the patches near the center are likely to belong to the target as opposed to other ones.

patches have dense representations (as illustrated in the top row of Figure 2). However, for a bad candidate, the local image patches have more dense coefficients, and the pooled features are smaller (as illustrated in the bottom row of Figure 2). To demonstrate the advantage of the proposed alignment-pooling algorithm, we compare the confidence map obtained by three kinds of pooling methods within a range around target. Based on these observations as illustrated in Figure 3, accurate localization of the target object can be achieved by the proposed local sparse representation with alignment-pooling.

4. Template Update

Tracking with fixed templates is prone to fail in dynamic scenes as it does not consider inevitable appearance change due to factors such as illumination and pose change. However, if we update the template too frequently with new observations, errors are likely to accumulate and the tracker will drift away from the target. Numerous approaches have been proposed for template update [15, 18, 16]. Ross *et al*. [18] extend the sequential Karhunen-Loeve algorithm and propose a new incremental principal component analysis (PCA) algorithm to update both the eigenbasis and the mean as new observations arrive. However the PCA based representation is sensitive to partial occlusion because of the assumption that reconstruction error is Gaussian distributed with small variance. Mei and Ling [16, 17] apply sparse representation to visual tracking and employ both target templates and trivial templates to handle outliers and partial occlusion. However, this method is not equipped with any mechanism to handle the drifting problem. In this paper, we introduce subspace learning into sparse representation to adapt templates to the appearance change of the target, and reduce the influence of the occluded target template as well.

In many tracking methods, the earlier tracking results are more accurate so they should be stored longer than newly acquired results in the template stack. One way to balance between the old and new templates is to assign different update probability to the templates. We generate a cumulative probability sequence

$$L_p = \left\{0, \frac{1}{2^{n-1}-1}, \frac{3}{2^{n-1}-1}, \ldots, 1\right\}, \tag{4}$$

and generate a random number r according to uniform distribution on the unit interval $[0, 1]$. By determining which section the random number lies in, we can choose the template to be replaced. This leads to slow update of old templates and quick update of new ones, and thereby alleviating the drifting problem.

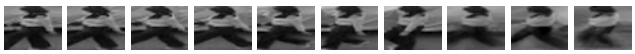
The strength of both sparse representation and subspace learning is exploited to model the updated template. We collect the tracking results of the target object and then carry out the incremental learning method proposed in [18]. Not only can this incremental method adapt to the appearance change but also preserve visual information the collected observations have in common. The estimated target can be modeled by a linear combination of the PCA basis vectors and additional trivial templates employed in [16]

$$\mathbf{p} = \mathbf{U}\mathbf{q} + \mathbf{e} = \left[\begin{array}{cc} \mathbf{U} & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \mathbf{q} \\ \mathbf{e} \end{array}\right], \tag{5}$$

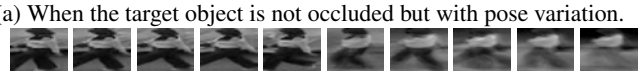
where $\mathbf{p}$ denotes the observation vector, $\mathbf{U}$ is the matrix composed of eigenbasis vectors, $\mathbf{q}$ is the coefficients of eigenbasis vectors and $\mathbf{e}$ indicates the pixels in $\mathbf{p}$ that are corrupted or occluded. As the error caused by occlusion and noise is arbitrary and sparse, we solve the problem as ℓ_1 regularized least square problem,

$$\min_{\mathbf{c}} \|\mathbf{p} - \mathbf{H}\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_1, \tag{6}$$

where $\mathbf{H} = \left[\begin{array}{cc} \mathbf{U} & \mathbf{I} \end{array}\right]$, $\mathbf{c} = \left[\begin{array}{cc} \mathbf{q} & \mathbf{e} \end{array}\right]^\top$ and λ is the regularization parameter. The coefficients of trivial templates are employed to account for noise or occlusion and avoid much occlusion to be updated into the template set. Thus the reconstructed image using only PCA basis vectors is not sensitive to the influence of occlusion. The reconstructed image is then used for updating the template to be replaced. This process can be viewed as introducing sparsity into subspace representation. Some templates obtained from the



(a) When the target object is not occluded but with pose variation.



(b) When the target is occluded.

Figure 4. Examples of templates obtained by the proposed template update strategy.

above-mentioned process are shown in Figure 4. We can see that the templates obtained when no occlusion occurs can adapt to the appearance change of the target. When there is occlusion, the templates focus on the parts which are not contaminated. With this template update strategy, our method can adapt to the appearance change of the target and handle the partial occlusion as well. The template update strategy is summarized in Algorithm 1.

Algorithm 1: Template Update

Input: Observation vector of target estimation $\mathbf{p}$, eigenbasis vectors $\mathbf{U}$, template set $\mathbf{T}$ and regularization parameter λ

1: Generate a sequence of number in ascending order and normalize them into $[0, 1]$ as the probability for template update

2: Generate a random number between 0 and 1 which is for the selection of which template to be discarded

3: Solve **Eq.** 6 and obtain $\mathbf{q}$ and $\mathbf{e}$

4: Add $\hat{\mathbf{p}} = \mathbf{U}\mathbf{q}$ to the end of the template set $\mathbf{T}$

Output: New template set $\mathbf{T}$

5. Proposed Tracking Algorithm

In this paper, object tracking is carried out within the Bayesian inference framework. Given the observation set of target $z_{1:t} = \{z_1, \ldots, z_t\}$ up to the t-th frame, the target state variable x_t can be computed by the maximum a posteriori estimation,

$$\hat{x}_t = \arg\max_{x_t^i} p\left(x_t^i | z_{1:t}\right), \tag{7}$$

where x_t^i indicates the state of the i-th sample. The posterior probability $p\left(x_t|z_{1:t}\right)$ can be inferred by the Bayesian theorem recursively,

$$p\left(x_t|z_{1:t}\right) \propto p\left(z_t|x_t\right) \int p\left(x_t|x_{t-1}\right) p\left(x_{t-1}|z_{1:t-1}\right) dx_{t-1}, \tag{8}$$

where $p\left(x_t|x_{t-1}\right)$ denotes the dynamic model and $p\left(z_t|x_t\right)$ denotes the observation model. The dynamic model $p\left(x_t|x_{t-1}\right)$ describes the temporal correlation of the target states between consecutive frames. We apply the affine transformation with six parameters to model the target motion between two consecutive frames. The state transition is formulated as $p\left(x_t|x_{t-1}\right) = N\left(x_t; x_{t-1}, \Sigma\right)$, where Σ is a diagonal covariance matrix whose elements are the variances of the affine parameters.

The observation model $p\left(z_t|x_t\right)$ denotes the likelihood of the observation z_t at state x_t. It plays an important role in robust tracking. In our method, the observation model is constructed by

$$p\left(z_t|x_t\right) \propto \sum_{k=1}^{N} \mathbf{f}_{\mathbf{k}}, \tag{9}$$

where the right side of the equation denotes the similarity between the candidate and the target based on the pooled feature $\mathbf{f}$. With the template updated incrementally, the observation model is able to adapt to the appearance change of the target.

6. Experiments

The proposed algorithm is implemented in MATLAB and runs at 1.5 frames per second on a Pentium 2.7 GHz Dual Core PC with 2GB memory. The ℓ_1 minimization problem is solved with the SPAMS package [14] and the regularization constant λ is set to 0.01 in all experiments. For each sequence, the location of the target object is manually labeled in the first frame. We resize the target image patch to 32×32 pixels and extract overlapped 16×16 local patches within the target region with 8 pixels as step length. As for the template update, 8 eigenvectors are used to carry out incremental subspace learning method in all experiments every 5 frames. The MATLAB source codes and datasets are available on our websites (`http://ice.dlut.edu.cn/lu/publications.html`, `http://faculty.ucmerced.edu/mhyang/pubs.html`).

We evaluate the performance of the proposed algorithm on nine challenging sequences from prior work [1, 3, 10, 18, 19], the CAVIAR data set (`http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/`) and our own. The challenges of these videos include illumination variation, partial occlusion, pose variation,background clutter and scale change. The proposed approach is compared with six state-of-the-art tracking methods including incremental visual tracking (IVT) method [18], fragment-based (FragTrack) tracking method [1], ℓ_1 tracker (ℓ_1) [16], multiple instance learning (MIL) tracker [3], visual tracking decomposition (VTD) method [10] and P-N learning (PN) tracker [9]. For fair evaluation, we evaluate the proposed tracker against those methods using the source codes provided by the authors. Each tracker is run with adjusted parameters.

6.1. Quantitative Evaluation

Two evaluation criteria are employed to quantitatively assess the performance of the trackers. Figure 5 presents the relative position errors (in pixels) between the center and the tracking results. Table 1 summarizes the average

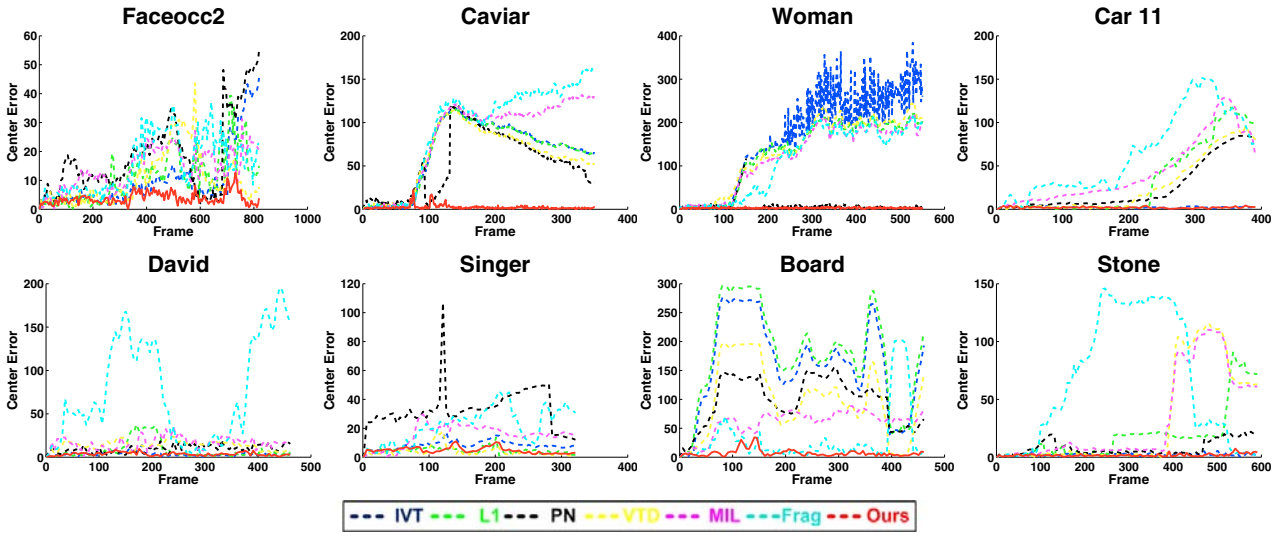


Figure 5. Quantitative evaluation of the trackers in terms of position errors (in pixels).

center location errors in pixels. In addition, given the tracking result R_T and the ground truth R_G, we use the detection criterion in the PASCAL VOC [6] challenge, i.e., $score = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$ to evaluate the success rate. Table 2 gives the average success rates. Overall, the proposed tracker performs favorably against state-of-the-art methods. The performance of our approach can be attributed to the efficient pooling methods across sparse codes of local image patches with a spatial layout.

	IVT	ℓ_1	PN	VTD	MIL	FragTrack	Ours
Faceocc2	**10.2**	11.1	18.6	10.4	14.1	15.5	**3.8**
Caviar	66.2	65.9	**53.0**	60.9	83.9	94.2	**2.3**
Woman	167.5	131.6	**9.0**	136.6	122.4	113.6	**2.8**
Car 11	**2.1**	33.3	25.1	27.1	43.5	63.9	**2.0**
David	**3.6**	7.6	9.7	13.6	16.1	76.7	**3.6**
Singer	8.5	**4.6**	32.7	**4.1**	15.2	22.0	4.8
Board	165.4	177.0	97.3	96.1	60.1	**31.9**	**7.3**
Stone	**2.2**	19.2	8.0	31.4	32.3	65.9	**1.8**

Table 1. Average center error (in pixels). The best two results are shown in red and blue fonts.

	IVT	ℓ_1	PN	VTD	MIL	FragTrack	Ours
Faceocc2	0.59	**0.67**	0.49	0.59	0.61	0.60	**0.82**
Caviar	**0.21**	0.20	**0.21**	0.19	0.19	0.19	**0.84**
Woman	0.19	0.18	**0.60**	0.15	0.16	0.20	**0.78**
Car 11	**0.81**	0.44	0.38	0.43	0.17	0.09	**0.81**
David	**0.72**	0.63	0.60	0.53	0.45	0.19	**0.79**
Singer	0.66	0.70	0.41	**0.79**	0.33	0.34	**0.81**
board	0.17	0.15	0.31	0.36	0.51	**0.73**	**0.74**
Stone	**0.66**	0.29	0.41	0.42	0.32	0.15	**0.56**

Table 2. Success rate of tracking methods. The best two results are shown in red and blue fonts.

6.2. Qualitative Evaluation

Occlusion: Figure 6 demonstrates how the proposed method performs when the target undergoes heavy occlusion or long-time partial occlusion. In the *Faceocc2* sequence, numerous trackers drift away from the target or do not scale well when the face is heavily occluded. Our tracker is able to track the target accurately because the structural local sparse appearance model has both spatial and partial information of the target. Those information helps avoid much influence of occlusion and better estimate the target. In the *Caviar* sequence, numerous methods fail to track the target because there are similar objects around it when heavy occlusion occurs. Our tracker does not drift away when the target reappears again because it is easier to differentiate the target and similar objects using both holistic and local information. Furthermore, our tracker is not affected much by occlusion owing to the structural local sparse appearance model and robust template update scheme. In the *Woman* sequence, the target object undergoes pose variation together with long-time partial occlusion. Based on the local patches and adaptive template update strategy, our tracker focuses more on the upper body which remains almost the same though the lower body changes a lot or is heavily occluded. It can successfully track the target throughout the entire sequence. The PN tracker (based on object detection with global search) is able to re-acquire the target when the target object reappears after occlusion. However, the other trackers lock on a car with similar color to the trousers when the legs of the woman are heavily occluded.

Illumination change: Figure 7 presents the tracking results in the sequences with large illumination variation. In the *Car 11* sequence, the contrast between the target and the background is low. The IVT tracker and our method perform well in tracking the vehicle while the other methods drift to the cluttered background or other vehicles when drastic illumination variation occurs. This can be attributed

Figure 6. Tracking results when the target objects are heavily occluded.

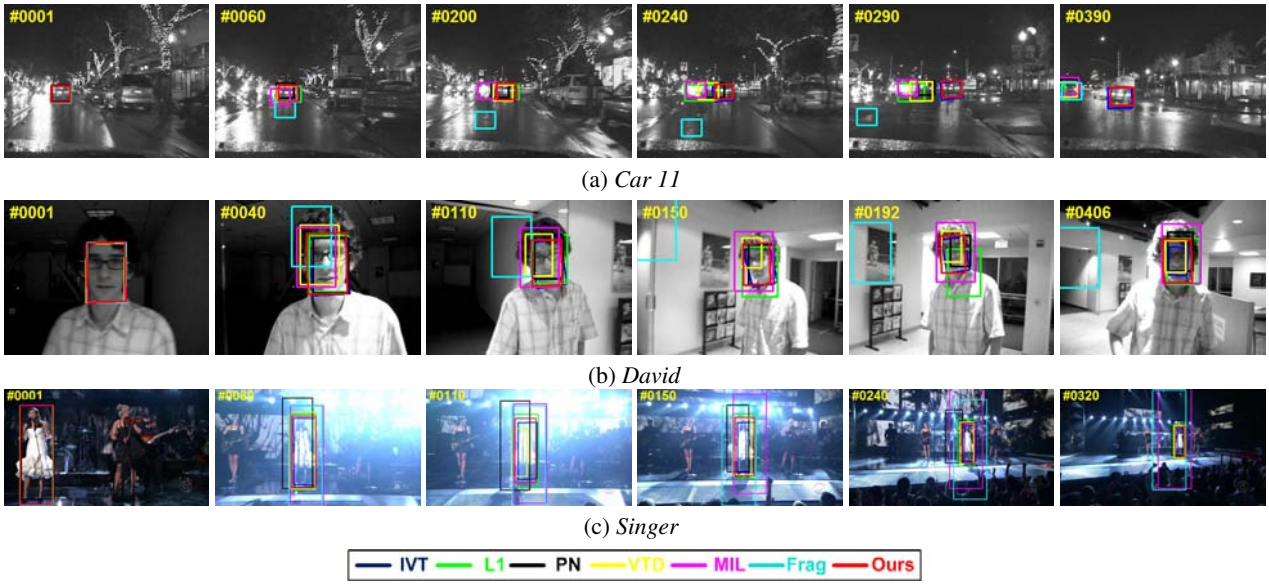
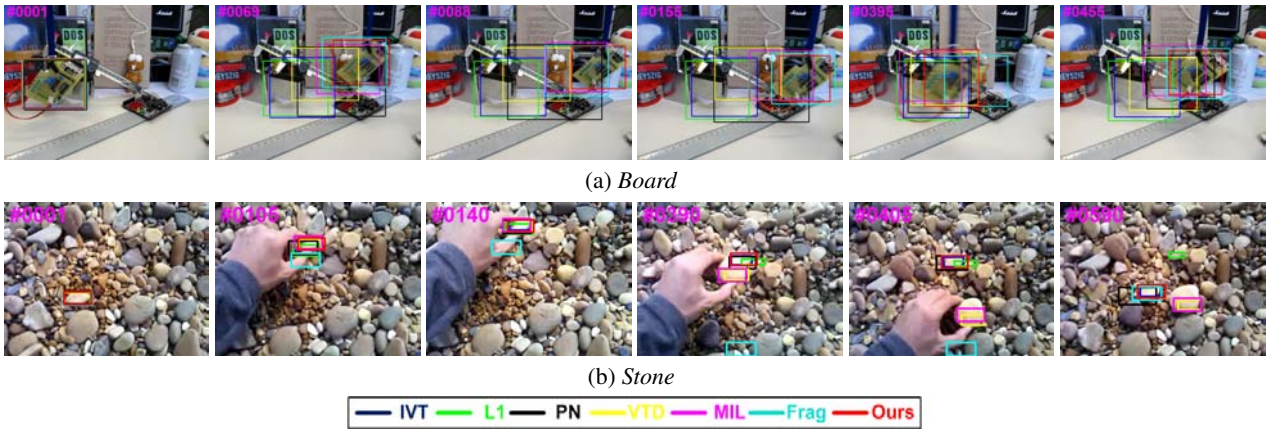


Figure 7. Tracking results when there is large illumination variation.

to the use of incremental subspace learning which is able to capture appearance change due to lighting change. In the *David* sequence, a person walks out of the dark conference room and into an area with spot lights. Likewise, in the *Singer* sequence a woman undergoes large appearance change due to drastic illumination variation and scale change. While a few trackers are able to keep track of the target to the end, the proposed algorithm achieves low tracking error and high success rate.

Background clutter: Figure 8 presents the tracking results where the target objects appear in background clutters. For *Board* sequence, most trackers drift away from the target as holistic representations are not effective in handling objects with large shape variations. The FragTrack and proposed methods are able to track the target better due to the use of local appearance models. The *Stone* sequence is challenging as there there are numerous stones of different shape and color. The FragTrack, MIL and VTD trackers drift to stones when the target is occluded whereas the IVT tracker and our



(a) *Board*

(b) *Stone*

Figure 8. Tracking results when the targets appear in cluttered backgrounds.

method successfully keep track of the target throughout the sequence. The PN tracker (based on object detection with global search) is able to re-acquire the target again after drifting to the background, but with higher tracking errors and lower success rate.

7. Conclusion

In this paper, we propose an efficient tracking algorithm based on structural local sparse appearance model and adaptive template update strategy. The proposed method exploits both spatial and local information of the target by alignment-pooling across the local patches with a spatial layout. This helps locate the target more accurately and is less insensitive to occlusion. In addition, sparse representation is combined with incremental subspace learning for template update. It not only adapts the tracker to account for appearance change of the target but also prevents incorrectly estimated or occluded observations from being put into the template set for update. Experimental results compared with several state-of-the-art methods on challenging sequences demonstrate the effectiveness and robustness of the proposed algorithm.

Acknowledgements

X. Jia and H. Lu are supported by the National Natural Science Foundation of China #61071209. M.-H. Yang is supported by the US National Science Foundation CAREER Grant #1149783 and IIS Grant #1152576.

References

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

[2] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261, 2007.

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

[4] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):163–84, 1998.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.

[6] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[7] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, 2006.

[8] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008.

[9] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[10] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010.

[11] B. Liu, J. Huang, L. Yang, and C. A. Kulikowski. Robust tracking using local sarse appearance model and k-selection. In *CVPR*, 2011.

[12] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. A. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, 2010.

[13] B. D. Lucas and T. Kanade. An iterative image registration techneque with an application to stereo vision. In *IJCAI*, 1981.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[15] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26:810–815, 2004.

[16] X. Mei and H. Ling. Robust visual tracking using L1 minimization. In *ICCV*, 2009.

[17] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient L1 tracker with occlusion detection. In *CVPR*, 2011.

[18] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008.

[19] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *CVPR*, 2010.

[20] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011.

[21] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[22] J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.