Robust Visual Tracking using ℓ_1 Minimization

Xue Mei
Center for automation Research
Electrical & Computer Engineering Department
University of Maryland, College Park, MD, USA
xuemei@umiacs.umd.edu

Haibin Ling
Center for Information Science & Technology
Computer & Information Science Department
Temple University, Philadelphia, PA, USA
hbling@temple.edu

Abstract

In this paper we propose a robust visual tracking method by casting tracking as a sparse approximation problem in a particle filter framework. In this framework, occlusion, corruption and other challenging issues are addressed seamlessly through a set of trivial templates. Specifically, to find the tracking target at a new frame, each target candidate is sparsely represented in the space spanned by target templates and trivial templates. The sparsity is achieved by solving an ℓ_1-regularized least squares problem. Then the candidate with the smallest projection error is taken as the tracking target. After that, tracking is continued using a Bayesian state inference framework in which a particle filter is used for propagating sample distributions over time. Two additional components further improve the robustness of our approach: 1) the nonnegativity constraints that help filter out clutter that is similar to tracked targets in reversed intensity patterns, and 2) a dynamic template update scheme that keeps track of the most representative templates throughout the tracking procedure. We test the proposed approach on five challenging sequences involving heavy occlusions, drastic illumination changes, and large pose variations. The proposed approach shows excellent performance in comparison with three previously proposed trackers.

1. Introduction

Visual tracking is a critical task in many computer vision applications such as surveillance, robotics, human computer interaction, vehicle tracking, and medical imaging, etc. The challenges in designing a robust visual tracking algorithm are caused by the presence of noise, occlusion, varying viewpoints, background clutter, and illumination changes [32]. A variety of tracking algorithms have been proposed to overcome these difficulties.

In this paper, we develop a robust visual tracking framework by casting the tracking problem as finding a sparse approximation in a template subspace. Motivated by the work in [30], we propose handling occlusion using trivial templates, such that each trivial template has only one nonzero element (see Fig. 1). Then, during tracking, a target candidate is represented as a linear combination of the template set composed of both target templates (obtained from previous frames) and trivial templates. The number of target templates are far fewer than the number of trivial templates. Intuitively, a good target candidate can be efficiently represented by the target templates. This leads to a sparse coefficient vector, since coefficients corresponding to trivial templates (named trivial coefficients) tend to be zeros. In the case of occlusion (and/or other unpleasant issues such as noise corruption or background clutter), a limited number of trivial coefficients will be activated, but the whole coefficient vector remains sparse. A bad target candidate, on the contrary, often leads to a dense representation[1] (e.g., Fig. 3). The sparse representation is achieved through solving an ℓ_1-regularized least squares problem, which can be done efficiently through convex optimization. Then the candidate with the smallest target template projection error is chosen as the tracking result. After that, tracking is led by the Bayesian state inference framework in which a particle filter is used for propagating sample distributions over time.

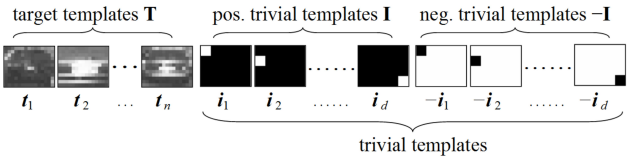

Figure 1. Templates used in our proposed approach (from the second testing sequence, Fig. 5).

Two additional components are included in our approach to further improve robustness. First, we enforce nonnegativity constraints to the sparse representation. These con-

[1] Candidates similar to trivial templates have sparse representations, but they are easily filtered out for their large dissimilarities to target templates.

straints are especially helpful to eliminate clutter that is similar to target templates with reversed intensity patterns. The constraints are implemented by including both positive and negative trivial templates in the template set. Second, we dynamically update the target template set to keep the representative templates throughout the tracking procedure. This is done by adjusting template weights by using the coefficients in the sparse representation. We tested the proposed approach on five video sequences involving heavy occlusion, large illumination and pose changes. The proposed approach shows excellent performance in comparison with three previously proposed trackers, including the Mean Shift (MS) tracker [9], the covariance (CV) tracker [27], and the appearance adaptive particle filter (AAPF) tracker [34].

The rest of the paper is organized as follows. In the next section related works are summarized. After that, the particle filter algorithm is reviewed in §2. §3 details the tracking algorithm using ℓ_1 minimization and our robust template update scheme. Experimental results on both methods are reported in §4. We conclude this paper in §5.

1.1. Related Work

Early works used the sum of squared difference (SSD) as a cost function in the tracking problem [3]. A gradient descent algorithm was most commonly used to find the minimum [16]. Subsequently, more robust similarity measures have been applied and the mean-shift algorithm or other optimization techniques utilized to find the optimal solution [9]. A mixture model of three components with an online EM algorithm is proposed to model the appearance variation during tracking [20]. The graph-cut algorithm was also used for tracking by computing an illumination invariant optical flow field [14]. The Covariance tracker [27] was proposed to successfully track nonrigid objects using a covariance based object description that fuses different types of features and modalities.

Tracking can be considered as an estimation of the state for a time series state space model. The problem is formulated in probabilistic terms. Early works uses a Kalman filter to provide solutions that are optimal for a linear Gaussian model. The particle filter, also known as the sequential Monte Carlo method [13], is one of the most popular approaches. It recursively constructs the posterior probability density function of the state space using Monte Carlo integration. It has been developed in the computer vision community and applied to tracking problems under the name Condensation [19]. In [34], an appearance-adaptive model is incorporated in a particle filter to realize robust visual tracking and classification algorithms. A hierarchical particle filter is used for multiple object tracking in [31].

Tracking can be considered as finding the minimum distance from the tracked object to the subspace represented by the training data or previous tracking results [4, 17]. In [28], the authors present a tracking method that incrementally learns a low-dimensional subspace representation, efficiently adapting online to changes in the appearance of the target. Tracking can also be considered as a classification problem and a classifier is trained to distinguish the object from the background [1, 7]. In [1], a feature vector is constructed for every pixel in the reference image and an adaptive ensemble of classifiers is trained to separate pixels that belong to the object from pixels that belong to the background. In [7], a confidence map is built by finding the most discriminative RGB color combination in each frame. A hybrid approach that combines a generative model and a discriminative classifier is used to capture appearance changes and allow reacquisition of an object after total occlusion [33]. Other studies on robust visual tracking can be found in [18, 35, 10, 29, 2], etc.

Our work is motivated by recent advances in sparse representation [5, 12] and its application in computer vision. The most relevant work is [30] where sparse representation is applied for robust face recognition. Other applications include background subtraction [6], media recovery [15], texture segmentation [24], and lighting estimation [26], etc.

2. Particle Filter

The particle filter [13] is a Bayesian sequential importance sampling technique for estimating the posterior distribution of state variables characterizing a dynamic system. It provides a convenient framework for estimating and propagating the posterior probability density function of state variables regardless of the underlying distribution. It consists of essentially two steps: prediction and update. Let x_t denote the state variable describing the affine motion parameters of an object at time t. The predicting distribution of x_t given all available observations $z_{1:t-1} = \{z_1, z_2, \cdots, z_{t-1}\}$ up to time $t-1$, denoted by $p(x_t|z_{1:t-1})$, is recursively computed as

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \quad (1)$$

At time t, the observation z_t is available and the state vector is updated using the Bayes rule

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad (2)$$

where $p(z_t|x_t)$ denotes the observation likelihood.

In the particle filter, the posterior $p(x_t|z_{1:t})$ is approximated by a finite set of N samples $\{x_t^i\}_{i=1,\cdots,N}$ with importance weights w_t^i. The candidate samples x_t^i are drawn from an importance distribution $q(x_t|x_{1:t-1}, z_{1:t})$ and the weights of the samples are updated as

$$w_t^i = w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t|x_{1:t-1}, z_{1:t})} \quad (3)$$

The samples are resampled to generate a set of equally weighted particles according to their importance weights to avoid degeneracy. In the case of the bootstrap filter $q(x_t|x_{1:t-1}, z_{1:t}) = p(x_t|x_{t-1})$ and the weights become the observation likelihood $p(z_t|x_t)$.

In the tracking framework, we apply an affine image warping to model the object motion of two consecutive frames. The state variable x_t is modeled by the six parameters of the affine transformation parameters $x_t = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_x, t_y)$, where $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are the deformation parameters and (t_x, t_y) are the 2D translation parameters. By applying an affine transformation using x_t as parameters, we crop the region of interest z_t from the image and normalize it to be the same size as the target templates in the gallery. We employ a Gaussian distribution to model the state transition distribution $p(x_t|x_{t-1})$. We also assume the six parameters of the affine transformation are independent. The observation model $p(z_t|x_t)$ reflects the similarity between a target candidate and the target templates. In this paper, $p(z_t|x_t)$ is formulated from the error approximated by the target templates using ℓ_1 minimization.

3. ℓ_1 Minimization Tracking

3.1. Sparse Representation of a Tracking Target

The global appearance of one object under different illumination and viewpoint conditions is known to lie approximately in a low dimensional subspace. Given target template set $\mathbf{T} = [\mathbf{t}_1 \dots \mathbf{t}_n] \in \mathbb{R}^{d \times n}$ ($d >> n$), containing n target templates such that each template $\mathbf{t}_i \in \mathbb{R}^d$ (we stack template image columns to form a 1D vector), a tracking result $\mathbf{y} \in \mathbb{R}^d$ approximately lies in the linear span of $\mathbf{T}$,

$$\mathbf{y} \approx \mathbf{Ta} = a_1\mathbf{t}_1 + a_2\mathbf{t}_2 + \cdots + a_n\mathbf{t}_n \ , \tag{4}$$

where $\mathbf{a} = (a_1, a_2, \cdots, a_n)^\top \in \mathbb{R}^n$ is called a *target coefficient vector*.

In many visual tracking scenarios, target objects are often corrupted by noise or partially occluded. The occlusion creates unpredictable errors. It may affect any part of the image and appear at any size on the image. To incorporate the effect of occlusion and noise, Equation 4 is rewritten as

$$\mathbf{y} = \mathbf{Ta} + \epsilon \ , \tag{5}$$

where ϵ is the error vector – a fraction of its entries are nonzero. The nonzero entries of ϵ indicate the pixels in $\mathbf{y}$ that are corrupted or occluded. The locations of corruption can differ for different tracking images and are unknown to the computer. Following the scheme in [30], we can use trivial templates $\mathbf{I} = [\mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_d] \in \mathbb{R}^{d \times d}$ to capture the occlusion as

$$\mathbf{y} = \begin{bmatrix}\mathbf{T}, & \mathbf{I}\end{bmatrix} \begin{bmatrix}\mathbf{a} \\ \mathbf{e}\end{bmatrix} \ , \tag{6}$$

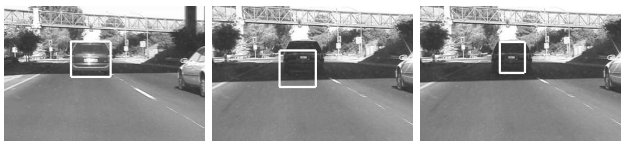

Figure 2. Left: target template. Middle: tracking result without non-negativity constraint. Right: tracking result with non-negativity constraint.

where a *trivial template* $\mathbf{i}_i \in \mathbb{R}^d$ is a vector with only one nonzero entry (i.e. $\mathbf{I}$ is an identity matrix), and $\mathbf{e} = (e_1, e_2, \cdots, e_d)^\top \in \mathbb{R}^d$ is called a *trivial coefficient vector*.

3.2. Nonnegativity Constraints

In principle, the coefficients in $\mathbf{a}$ can be any real numbers if the target templates are taken without restrictions. However, we argue that in tracking, a tracking target can almost always represented by the target templates dominated by nonnegative coefficients. Here by "dominated" we mean that the templates that are most similar to the tracking target are positively related to the target. This is true when we start tracking from the second frame (the target is selected in the first frame manually or by a detection method), the target in the second frame will look more like the target in the first frame such that the coefficient is positive when the target in the first frame is used to approximate the target in the second frame. In new frames the appearance of targets may change, but new templates will be brought in (may replace old templates) and the coefficients will still be positive for the most similar target templates in the following frames.

Another important argument for including nonnegative coefficients comes from their ability to filter out clutter that is similar to target templates at reversed intensity patterns, which often happens when shadows are involved. We give an example in Fig. 2. In Fig. 2, the left image shows the first frame in which the target template is created as in the bounding box. The middle image shows the tracking result without nonnegativity constraints, where the tracking result is way off the correct location. By checking the coefficients in $\mathbf{a}$, we found that the failure is because the intensity pattern in the tracking result (dark in top and light in bottom) is roughly reversed compared to the target template (dark in bottom and light in top). This problem can be avoided by enforcing the nonnegativity constraints, as shown in the right image.

Enforcing nonnegativity constraints on the target coefficient vector $\mathbf{a}$ is straightforward. However, it is unreasonable to put such constraints directly on the trivial coefficient vector $\mathbf{e}$. For this reason, we propose extending the trivial templates by including *negative trivial templates* as well. Consequently, model (6) is now written as

$$\mathbf{y} = \begin{bmatrix}\mathbf{T}, & \mathbf{I}, & -\mathbf{I}\end{bmatrix} \begin{bmatrix}\mathbf{a} \\ \mathbf{e}^+ \\ \mathbf{e}^-\end{bmatrix} \doteq \mathbf{Bc} \ , \quad \text{s.t. } \mathbf{c} \geq 0 \ , \tag{7}$$

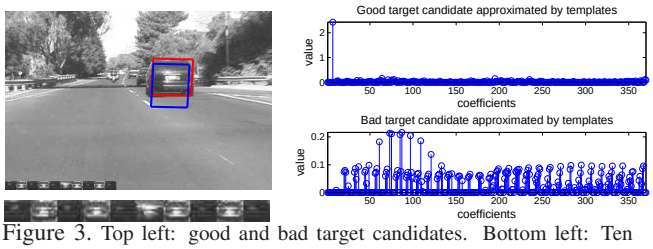

Figure 3. Top left: good and bad target candidates. Bottom left: Ten templates in the template set. They are the enlarged version of the templates shown on the bottom left corner of the top image. Top right: good target candidate approximated by template set. Bottom right: bad target candidate approximated by template set.

where $\mathbf{e}^+ \in \mathbb{R}^d, \mathbf{e}^- \in \mathbb{R}^d$ are called a *positive* trivial coefficient vector and a *negative* trivial coefficient vector respectively, $\mathbf{B} = [\mathbf{T}, \mathbf{I}, -\mathbf{I}] \in \mathbb{R}^{d\times(n+2d)}$, and $\mathbf{c}^\top = [\mathbf{a}, \mathbf{e}^+, \mathbf{e}^-] \in \mathbb{R}^{n+2d}$ is a non-negative coefficient vector. Example templates are illustrated in Fig. 1.

3.3. Achieving Sparseness through ℓ_1 Minimization

The system in (7) is underdetermined and does not have a unique solution for $\mathbf{c}$. The error caused by occlusion and noise typically corrupts a fraction of the image pixels. Therefore, for a good target candidate, there are only a limited number of nonzero coefficients in $\mathbf{e}^+$ and $\mathbf{e}^-$ that account for the noise and partial occlusion. Consequently, we want to have a sparse solution to (7). We exploit the compressibility in the transform domain by solving the problem as an ℓ_1-regularized least squares problem, which is known to typically yield sparse solutions [30]

$$\min ||\mathbf{Bc} - \mathbf{y}||_2^2 + \lambda||\mathbf{c}||_1 \ , \tag{8}$$

where $||.||_1$ and $||.||_2$ denote the ℓ_1 and ℓ_2 norms respectively.

Fig. 3 shows the coefficients approximated by the template set for the good and bad target candidates. The good and bad target candidates are shown in the red and blue bounding boxes on the top left image. The 10 target templates with size of 12×15 are shown on the left corner, while the enlarged version is shown on the bottom left image. The images on the right show the good and bad target candidate approximated by the template set, respectively. The first 10 coefficients correspond to the 10 target templates used in the tracking and the rest 360 coefficients correspond to the trivial templates. In the top right image, the seventh coefficient is relatively large comparing to the rest coefficients. Thus the seventh target template represents the good candidate well and the trivial templates have a small factor in approximating the good candidate. While in the bottom right image, the coefficients are densely populated and trivial templates account for most of the approximation for bad candidate in the left image.

Our implementation solves the ℓ_1-regularized least squares problem via an interior-point method based on [22].

Algorithm 1 Template Update

$\mathbf{y}$ is the newly chosen tracking target.
$\mathbf{a}$ is the solution to (8).
$\mathbf{w}$ is current weights, such that $w_i \leftarrow ||\mathbf{t}_i||_2$.
τ is a predefined threshold.
Update weights according to the coefficients of the target templates. $w_i \leftarrow w_i * exp(a_i)$.
if ($sim(\mathbf{y}, \mathbf{t}_m) < \tau$), where sim is a similarity function. It can be the angle between two vectors or SSD between two vectors after normalization. $\mathbf{t}_m$ has the largest coefficient a_m, that is, $m = arg\max_{1\le i\le n} a_i$ **then**
    $i_0 \leftarrow \arg\min_{1\le i\le n} w_i$
    $\mathbf{t}_{i_0} \leftarrow \mathbf{y}$, /*replace an old template*/.
    $w_{i_0} \leftarrow$ median($\mathbf{w}$), /*replace an old weight*/.
end if
Normalize $\mathbf{w}$ such that $sum(\mathbf{w}) = 1$.
Adjust $\mathbf{w}$ such that $max(\mathbf{w}) = 0.3$ to prevent skewing.
Normalize $\mathbf{t}_i$ such that $||\mathbf{t}_i||_2 = w_i$.

The method uses the preconditioned conjugate gradients (PCG) algorithm to compute the search direction and the run time is determined by the product of the total number of PCG steps required over all iterations and the cost of a PCG step. We use the code from [11] for the minimization task.

We then find the tracking result by finding the smallest residual after projecting on the target template subspace, i.e., $||\mathbf{y} - \mathbf{Ta}||_2$. Therefore, the tracking result is the sample of states that obtains the largest probability, that is, the smallest error.

3.4. Template Update

Template tracking was suggested in the computer vision literature in [23], dating back to 1981. The object is tracked through the video by extracting a template from the first frame and finding the object of interest in successive frames. A fixed appearance template is not sufficient to handle recent changes in the video, while a rapidly changing model is susceptible to drift. Approaches have been proposed to overcome the drift problem [25, 21] in different ways.

Intuitively, object appearance remains the same only for a certain period of time, but eventually the template is no longer an accurate model of the object appearance. If we do not update the template, the template cannot capture the appearance variations due to illumination or pose changes. If we update the template too often, small errors are introduced each time the template is updated. The errors are accumulated and the tracker drifts from the target. We tackle this problem by dynamically updating the target template set $\mathbf{T}$.

One important feature for ℓ_1 minimization is that it favors the template with larger norm because of the regular-

ization part $||\mathbf{c}||_1$. The larger norm of $\mathbf{t}_i$ is, the smaller coefficient a_i is needed in the approximation $||\mathbf{y}-\mathbf{Bc}||_2$. We exploit the characteristic by introducing a weight $w_i = ||\mathbf{t}_i||_2$ associated with each template $\mathbf{t}_i$. Intuitively, the larger the weight is, the more important the template is. At initialization, the first target template is manually selected from the first frame and applied zero-mean-unit-norm normalization. The rest target templates are created by perturbating one pixel in four possible directions at the corner points of the first template in the first frame. Thus we create all the target templates (10 for our experiments) at the first frame. The target template set $\mathbf{T}$ is then updated with respect to the coefficients of the tracking result.

The updating in our approach includes three operations: template replacement, template updating, and weight updating. If the tracking result $\mathbf{y}$ is not similar to the current template set $\mathbf{T}$, it will replace the least important template in $\mathbf{T}$ and be initialized to have the median weight of the current templates. The weight of each template increases when the appearance of the tracking result and template is close enough and decreases otherwise. The template update scheme is summarized in Algorithm 1.

4. Experiments

In order to evaluate the performance of the proposed tracking framework, five videos are used in the experiments. The first two videos consist of 8-bit gray scale images while the last three are composed of 24-bit color images. In our experiments, we compare the tracking results of our proposed method with those of state-of-the-art standard Mean Shift (MS) tracker [9], covariance (CV) tracker [27], and appearance adaptive particle filter (AAPF) tracker [34].

The first test sequence is an infrared (IR) image sequence from VIVID benchmark dataset [8] PkTest02. Some samples of the final tracking results are demonstrated in Fig. 4, where rows 1, 2, 3, and 4 are for our proposed tracker, MS tracker, CV tracker, and AAPF tracker, respectively, in which six representative frames of the video sequences are shown. The frame indexes are 1117, 1157, 1173, 1254, 1273, 1386. The target-to-background contrast is very low and the noise level is high for these IR frames. From Fig. 4, we see that our tracker is capable of tracking the object all the time even with severe occlusions by the trees on the roadside. In comparison, MS tracker locks onto the car behind the target. It keeps tracking it in the rest of the sequences and is unable to recover it. CV tracker fails to track the target in the fourth index frame similar to MS tracker, but it is able to recover the failure and track the target properly from the fifth index frame. In comparison, our proposed method avoids this problem and is effective under low contrast and noisy situation. AAPF achieves similar results to our method.

The second test sequence is obtained from http://www.cs.toronto.edu/~dross/ivt/. The vehicle undergoes drastic illumination changes as it passes beneath a bridge and under trees. Some samples of the final tracking results are demonstrated in Fig. 5, where rows 1, 2, 3, and 4 are for our proposed tracker, MS tracker, CV tracker, and AAPF tracker, respectively. The frame indexes are 181, 196, 233, 280, 308, 315. MS tracker loses the target very quickly and goes out of range from the fourth frame. CV tracker loses the target quickly and gets stuck on the background. Our tracker and AAPF are able to track the target well even though the illumination changes.

The third test sequence is obtained from http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/. In this video, the background color is similar to the color of the woman's trousers, and the man's shirt and pants have a similar color to the woman's coat. In addition, the woman undergoes partial occlusion. Some tracking result frames are given in Fig. 6. The frame indexes are 137, 183, 207, 225, 245, 271. It can be observed that the other trackers except our method start tracking the man when the woman is partially occluded at frame 207. Compared with other trackers, our tracker is more robust to the occlusion, which makes the target model not easily degraded by the outliers.

The fourth test sequence is obtained from http://vision.stanford.edu/~birch/headtracker/seq/. We show some samples of the tracking results for the trackers in Fig. 7. The six representative frame indexes are 422, 436, 448, 459, 466, and 474. The man's face is passing in front of the woman's face. Again, our method obtains good tracking results. The same for the MS and CV tracker. The AAPF tracker drifts apart when the severe occlusion happens in second frame.

The fifth test sequence is an airborne car video. The car is running on a curved road and passing beneath the trees. It undergoes heavy occlusions and large pose changes. We show some samples of the tracking results for the trackers in Fig. 8. The six representative frames indexes are 215, 331, 348, 375, 393, and 421. MS tracker loses the target very quickly and goes out of range in the sixth frame. Although CV and AAPF can track the target, they do not locate the target well. Our tracker tracks the target very well throughout the whole sequence.

5. Conclusion

In this paper we propose using a sparse representation for robust visual tracking. We model tracking as a sparse approximation problem and solve it through an ℓ_1-regularized least squares approach. For further robustness, we introduce nonnegativity constraints and dynamic template updating in our approach. In thorough experiments involving five challenging sequences and three other state-of-the-art trackers, our approach demonstrates very promising performance.

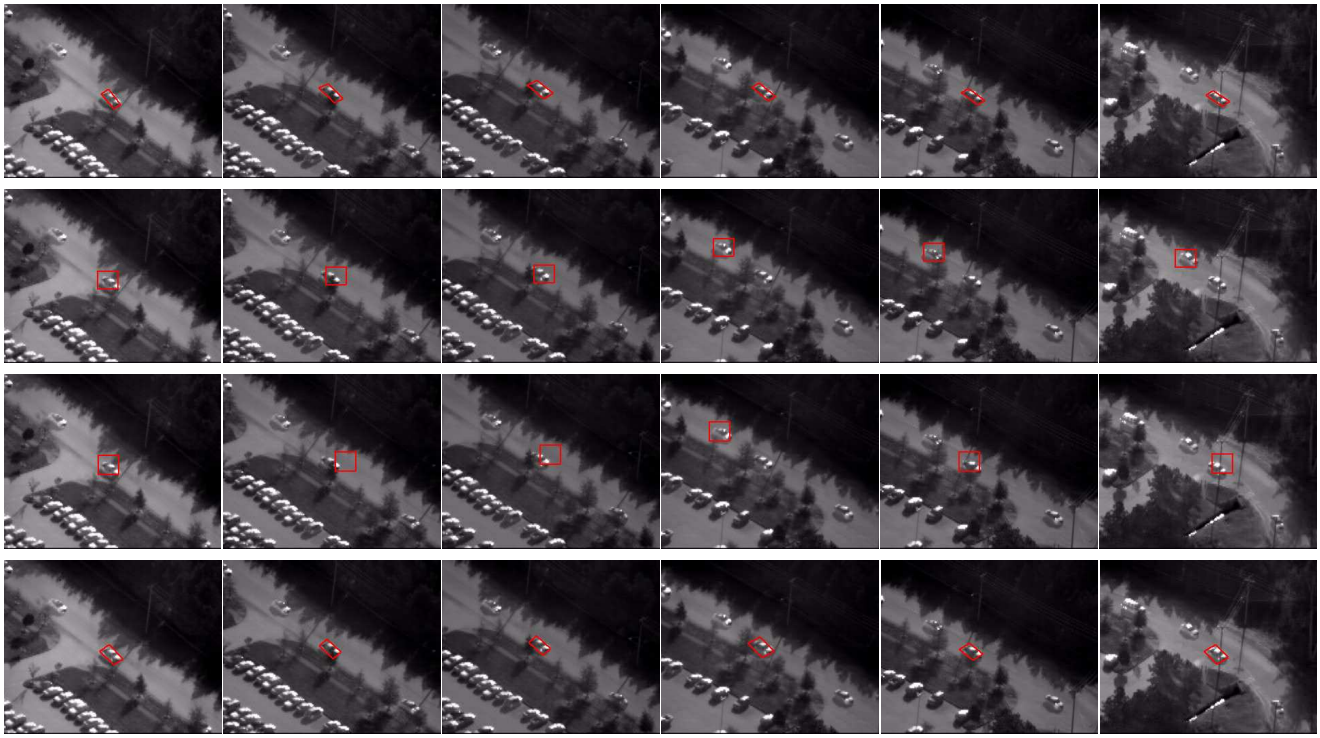

Figure 4. The tracking results of the first sequence: our proposed tracker (row 1), MS (row 2), CV (row 3), and AAPF (row 4) over representative frames with severe occlusion.

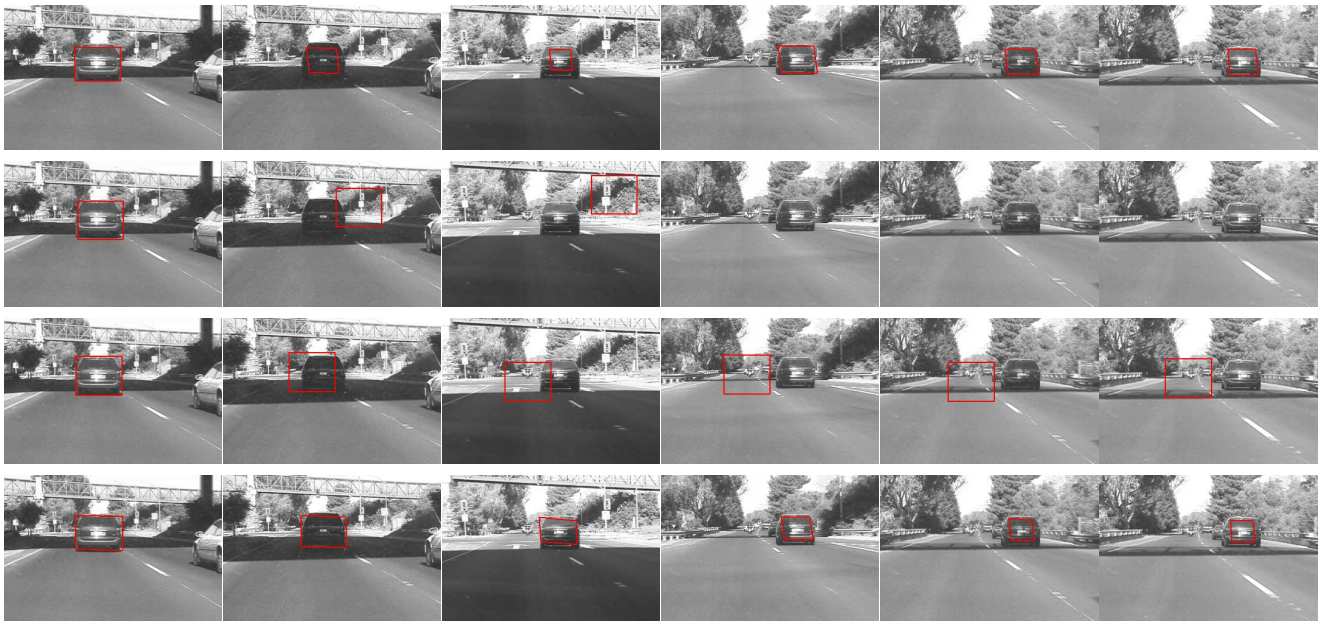

Figure 5. The tracking results of the second sequence: our proposed tracker (row 1), MS (row 2), CV (row 3), and AAPF (row 4) over representative frames with drastic illumination changes.

References

[1] S. Avidan. "Ensemble Tracking", *CVPR*, 494-501, 2005. 2

[2] B. Babenko, M. Yang, and S. Belongie. "Visual Tracking with Online Multiple Instance Learning", *CVPR*, 2009. 2

[3] S. Baker and I. Matthews. "Lucas-kanade 20 years on: A unifying framework", *IJCV*, 56:221-255, 2004. 2

[4] M. J. Black and A. D. Jepson. "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation", *IJCV*, 26:63-84, 1998. 2

Figure 6. The tracking results of the third sequence: our proposed tracker (row 1), MS (row 2), CV (row 3), and AAPF (row 4) over representative frames with partial occlusion and background clutter.

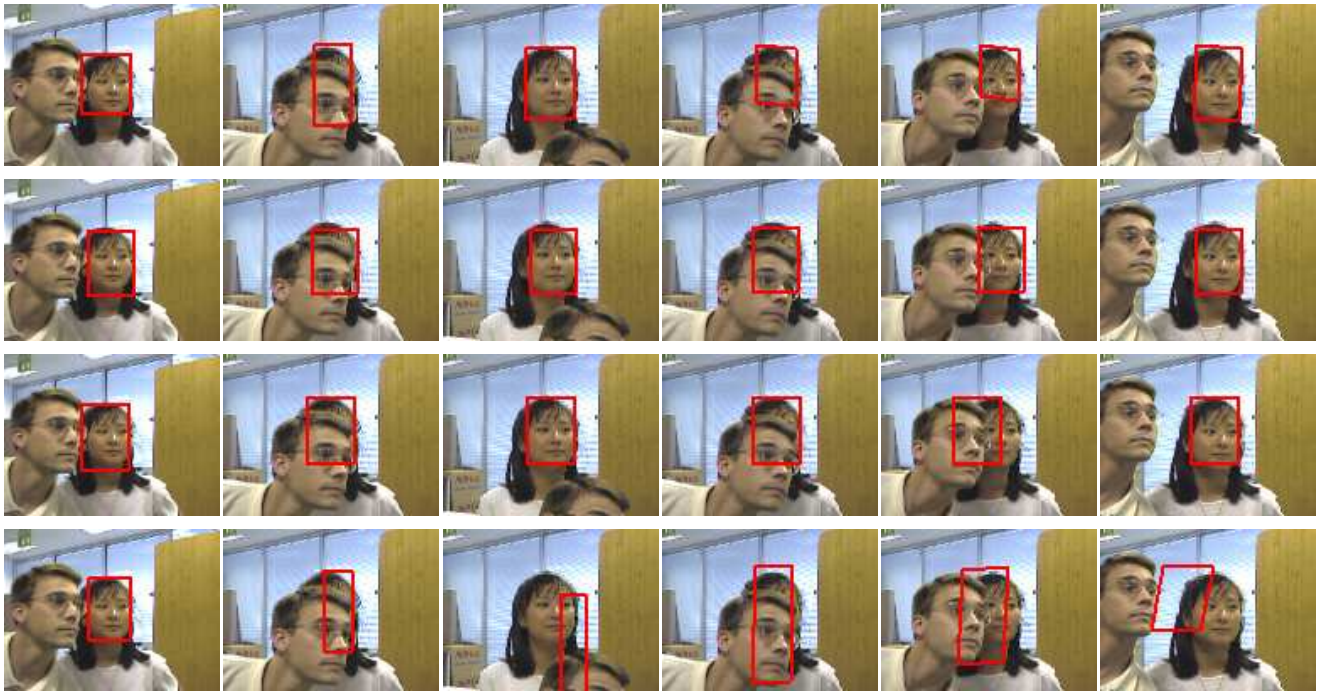

Figure 7. The tracking results of the fourth sequence: our proposed tracker (row 1), MS (row 2), CV (row 3), and AAPF (row 4) over representative frames with severe occlusion.

[5] E. Candès, J. Romberg, and T. Tao. "Stable signal recovery from incomplete and inaccurate measurements", *Comm. on Pure and Applied Math*, 59(8):1207-1223, 2006. 2

[6] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa. "Compressive Sensing for Background Subtraction", *ECCV*, 2008. 2

Figure 8. The tracking results of the fifth sequence: our proposed tracker (row 1), MS (row 2), CV (row 3), and AAPF (row 4) over representative frames with heavy occlusion and large pose variation.

[7] R. T. Collins and Y. Liu. "On-Line Selection of Discriminative Tracking Features", *ICCV*, 346-352, 2003. 2

[8] R. Collins, X. Zhou, and S. K. Teh. "An Open Source Tracking Testbed and Evaluation Web Site", *PETS*, 2005. 5

[9] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based object tracking", *PAMI*, 25:564-577, 2003. 2, 5

[10] D. Conte, P. Foggia, J.-M. Jolion, and M. Vento. "A graph-based, multi-resolution algorithm for tracking objects in presence of occlusions". *Pattern Recognition*, 39(4):562-572, 2006. 2

[11] *http://www.stanford.edu/˜boyd/l1_ls/*. 4

[12] D. Donoho. "Compressed Sensing", *IEEE Trans. Inf. Theory*, 52(4):1289-1306, 2006. 2

[13] A. Doucet, N. de Freitas, and N. Gordon. "Sequential Monte Carlo Methods in Practice". *Springer-Verlag*, 2001, New York. 2

[14] D. Freedman and M. W. Turek. "Illumination-Invariant Tracking via Graph Cuts", *CVPR*, 10-17, 2005. 2

[15] J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi. "Compressive Structured Light for Recovering Inhomogeneous Participating Media", *ECCV*, 2008. 2

[16] G. Hager and P. Belhumeur. "Real-time tracking of image regions with changes in geometry and illumination", *CVPR*, 403-410, 1996. 2

[17] J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman. "Visual tracking using learned subspaces", *CVPR*, 782-789, 2004. 2

[18] Y. Huang and I. A. Essa. "Tracking Multiple Objects through Occlusions". *CVPR* 2:1051-1058, 2005. 2

[19] M. Isard and A. Blake. "Condensation - conditional density propagation for visual tracking", *IJCV*, 29:5-28, 1998. 2

[20] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. "Robust online appearance models for visual tracking", *PAMI*, 25:1296-1311, 2003. 2

[21] T. Kaneko and O. Hori. "Feature Selection for Reliable Tracking using Template Matching", *CVPR*, 796-802, 2003. 4

[22] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. "A method for large-scale ℓ_1-regularized least squares", *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606-617, 2007. 4

[23] B. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision", *IJCAI*, 674-679, 1981. 4

[24] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. "Discriminative learned dictionaries for local image analysis", *CVPR*, 2008. 2

[25] I. Matthews, T. Ishikawa, and S. Baker. "The template update problem", *PAMI*, 810-815, 2004. 4

[26] X. Mei, H. Ling, and D.W. Jacobs. "Sparse Representation of Cast Shadows via ℓ_1-Regularized Least Squares", *ICCV*, 2009. 2

[27] F. Porikli, O. Tuzel and P. Meer. "Covariance tracking using model update based on lie algebra", *CVPR*, 728-735, 2006. 2, 5

[28] D. A. Ross, J. Lim, R. Lin and M. Yang. "Incremental learning for robust visual tracking", *IJCV*, 77:125-141, 2008. 2

[29] O. Williams, A. Blake, and R. Cipolla. "Sparse Bayesian Learning for Efficient visual Tracking", *PAMI*, 27:1292-1304, 2005. 2

[30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. "Robust Face Recognition via Sparse Representation", *PAMI*, 31(1):210-227, 2009. 1, 2, 3, 4

[31] C. Yang, R. Duraiswami, and L. S. Davis. "Fast multiple object tracking via a hierarchical particle filter", *ICCV*, 212-219, 2005. 2

[32] A. Yilmaz, O. Javed, and M. Shah. 'Object tracking: A survey". *ACM Comput. Surv.* 38(4), 2006. 1

[33] Q. Yu, T. B. Dinh and G. Medioni. "Online tracking and reacquistion using co-trained generative and discriminative trackers", *ECCV*, 678-691, 2008. 2

[34] S. K. Zhou, R. Chellappa, and B. Moghaddam. "Visual tracking and recognition using appearance-adaptive models in particle filters", *IEEE Trans. Image Processing*, 11:1491-1506, 2004. 2, 5

[35] Y. Zhou and H. Tao. "A background layer model for object tracking through occlusion." *ICCV*, 2003. 2