Online Discriminative Object Tracking with Local Sparse Representation

Qing Wang , Feng Chen, Wenli Xu
Automation, Tsinghua University
qing-wang07@mails.tsinghua.edu.cn

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

Abstract

We propose an online algorithm based on local sparse representation for robust object tracking. Local image patches of a target object are represented by their sparse codes with an over-complete dictionary constructed online, and a classifier is learned to discriminate the target from the background. To alleviate the visual drift problem often encountered in object tracking, a two-stage algorithm is proposed to exploit both the ground truth information of the first frame and observations obtained online. Different from recent discriminative tracking methods that use a pool of features or a set of boosted classifiers, the proposed algorithm learns sparse codes and a linear classifier directly from raw image patches. In contrast to recent sparse representation based tracking methods which encode holistic object appearance within a generative framework, the proposed algorithm employs a discrimination formulation which facilitates the tracking task in complex environments. Experiments on challenging sequences with evaluation of the state-of-the-art methods show effectiveness of the proposed algorithm.

1. Introduction

Object tracking has long been an important problem in computer vision which finds numerous applications in surveillance, human-computer interaction, vehicle navigation, to name a few. Although many tracking methods have been proposed and significant progress has been made within the last decades, this problem remains rather challenging. In order to develop robust object tracking algorithms, the main challenging issues including background clutter, illumination change, target pose change, occlusion, camera motion must be addressed.

In this paper, we propose an online tracking algorithm which does not assume any prior information of the target objects or the tracking scenario. Objects are represented with a novel local sparse representation and the tracking task is formulated as a classification problem with online update. For object representation, we first learn an over-complete dictionary with labeled data (i.e., detected or manually initialized target object) in the first frame and then represent each image patch inside the object region with its sparse code. Each sparse code is learned by simultaneously minimizing the reconstruction error and maximizing its sparsity of each image patch with an adaptive dictionary. An object is then represented by concatenating the sparse codes of all image patches. With this representation scheme, positive and negative samples are collected after the target object is labeled in the first frame, and a linear classifier is learned to separate the target from the background. Using the classification score as the likelihood of a test candidate belonging to the tracking object, the most likely target location in each frame can be determined. To account for the target and background appearance variations using dictionary and classifier update without introducing visual drift, we propose a two-stage tracking algorithm with particle filtering. For robust tracking, a static observation model and an adaptive observation model are exploited. The static observation model is constructed based on the initial dictionary and classifier obtained in the first frame whereas the adaptive observation model is constructed by the most recent dictionary and classifier. In each frame, samples are first processed using a particle filter with the adaptive observation model, and then further examined by a particle filter with the static observation model in order to determine the most likely target location. The dictionary and classifier are updated with image patches of the estimated target location.

Compared to existing algorithms for object tracking, the contributions of our method are as follows. First, we represent objects with sparse codes of local image patches for robust object tracking. In numerous vision problems, local descriptors have been shown to be more robust than the alternative holistic representations when objects undergo pose change, deformation and partial occlusion. By computing the sparse codes of all the gray-scale image patches inside an object and concatenating them together, an effective representation is obtained which facilitates the classifier to separate the foreground target from the cluttered background. Second, we formulate object tracking as a classification problem with sparse representation. All the recent

sparse representation based tracking algorithms [18, 16] are posed within the generative framework and use reconstruction errors to determine the likely locations of target objects. The proposed discriminative approach significantly facilitates the task in separating target objects from cluttered backgrounds. Different from most recent discriminative tracking methods which use multiple features or learn boosted classifiers [3, 9, 4], our algorithm learns one linear classifier based on local sparse representation with favorable tracking performance on challenging sequences. Third, we propose a simple but effective tracking method that alleviates the drift problem with adaptive dictionary and classifier to reflect appearance change of the target and background. The effectiveness of the proposed method are born out by experiments on several challenging sequences and quantitative evaluations with the state-of-the-art methods.

2. Related Work

There is a rich literature in object tracking, and existing tracking algorithms can be roughly categorized as either generative or discriminative approaches. To deal with the challenges mentioned above, most recent tracking algorithms focus on robust object representation schemes with generative appearance models and sophisticated classifiers.

Generative methods represent objects with models that have minimum reconstruction errors, and track targets by searching for the region most similar to the models in an image frame. To deal with the above-mentioned challenges in object tracking, most recent generative methods learn and maintain static or online appearance models. Black et al. [5] learn a subspace model offline to represent target objects at predefined views and build on the optical flow framework for tracking. In [6], Black et al. extend their subspace representation method to a mixture model which can better account for change of object appearance. To handle target appearance variations during tracking, Jepson et al. [12] learn a Gaussian mixture model of pixels to represent objects via an online expectation maximization (EM) algorithm. Instead of describing objects with a blob of pixels, David et al. [19] learn an adaptive linear subspace online for modeling target appearance and implement tracking with a particle filter. The recent development of sparse representation [20] has attracted considerable interest in object tracking [18, 16] due to its robustness to occlusion and image noise. As these methods exploit only generative representations of target objects and do not take the background into account, they are less effective for tracking in cluttered environments.

Discriminative methods pose object tracking as a binary classification problem in which the task is to distinguish the target region from the background in each image with samples drawn within a local regions from previous location. Contrasted to generative methods which only model the target appearance, discriminative algorithms use information from both the target and the background. Avidan [2] trains a Support Vector Machine (SVM) classifier offline and extends it within the optical flow framework for object tracking. Collins et al. [7] use variance ratio of foreground and background classes to determine discriminative features for object tracking. In [3], an ensemble tracking method is proposed in which a set of weak classifiers are trained and combined for distinguishing the target object and the background. The online boosting algorithm has also been used to select discriminative features for tracking [9].

To deal with the drift problem when updating the learned appearance models or online classifiers with newly obtained tracking results, numerous approaches have been proposed in recent years. Matthews et al. [17] propose an update method with the Lucas-Kanade algorithm by applying a template extracted in the most recent frame to estimate the tracking result first and then using the template from the first frame to determine the target location. In addition to supervised approaches for discriminative object tracking, Grabner et al. [10] treat all visual information from tracking results as unlabeled data and adapt a classifier within the semi-supervised learning framework. Babenko et al. [4] use multiple instance learning (MIL) to handle ambiguously labeled positive and negative data obtained online to reduce visual drifts. Kalal et al. [13] also regard tracking results from a classifier as unlabeled and exploit their underlying structure to select positive and negative samples for update.

3. Learning Representation and Classifier

In this paper, we use local sparse codes to represent objects and formulate tracking as a binary classification problem. We initialize the dictionary and classifier in the first frame after the target object is labeled manually or automatically. Both the dictionary and classifier are updated when new tracking results are available.

3.1. Object Representation by Local Sparse Coding

To generate an effective object representation, we first encode the local patches inside an object region using an over-complete dictionary and then aggregate the corresponding sparse codes. Although there are efficient algorithms [15, 21] for learning an over-complete dictionary, it is difficult to collect a sufficient number of training data only from the first tracking frame. In [18], perturbation around the target location is carried out for collecting multiple holistic target templates to construct the dictionary in the first frame. However, this method inevitably introduces some alignment errors or noise to the training data and consequently affects the learned basis of the dictionary. In this work, We use a different method to construct the dictionary. With the overlapped image patches extracted from the target object region in the first frame, we obtain the target basis

set $T = [\mathbf{t}_1, \ldots, \mathbf{t}_n] \in \mathbb{R}^{d \times n}$ by normalizing the vectorized image patches with zero mean and unit variance, where d is the dimensionality of the image vectors and n is the number of image vectors. The over-complete dictionary is constructed by

$$D = [T, I, -I], \tag{1}$$

where $I \in \mathbb{R}^{d \times d}$ is an identity matrix whose columns are the trivial bases for the dictionary. Similar to [18], the use of I and $-I$ maintains a non-negativity constraint and a sparsity constraint of the sparse codes when representing an image patch with the dictionary $D \in \mathbb{R}^{d \times (n+2d)}$. With our formulation, no initialization errors or noise are introduced into the dictionary.

With the dictionary D, we first encode the image patches inside the target object. Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ denote the vectorized image patches extracted from an object image, the sparse code $\mathbf{a}_i \in \mathbb{R}^{(n+2d)}$ corresponding to $\mathbf{x}_i$ is computed by:

$$\min_{\mathbf{a}_i} \tfrac{1}{2}\|\mathbf{x}_i - D\mathbf{a}_i\|_2^2 + \lambda_1 \|\mathbf{a}_i\|_1 + \tfrac{\lambda_2}{2}\|\mathbf{a}_i\|_2^2, \tag{2}$$

where λ_1 and λ_2 are regularization parameters. When $\lambda_2 = 0$, it leads to the ℓ_1-norm sparse coding problem which has been widely used [20, 18, 16]. The choice of the $\lambda_2 > 0$ makes the optimization problem strictly convex [22].

When the sparse codes $[\mathbf{a}_1, \ldots, \mathbf{a}_N]$ of all the image patches from an object region are computed, we aggregate them to obtain the object representation for visual tracking. There exist numerous methods for representing an object with a set of descriptors. Here we directly concatenate all these sparse codes together to represent the object, i.e., $\mathbf{z} = [\mathbf{a}_1^\top, \ldots, \mathbf{a}_N^\top]^\top$.

3.2. Classifier Learning with Sparse Representation

We pose visual tracking as a classification problem, i.e., a problem in which the aim is to separate the target object from the background. With our object representation, image patches from the target and the background can be represented by different bases in the dictionary. Using the proposed dictionary, the image patches from a target object are likely to be well reconstructed by only the target basis set T, but image patches from the background may need trivial bases for good reconstruction. Therefore, it is easier to separate the target object from the background with our sparse representation than using the raw image features. Different from the recent discriminative tracking algorithms which use boosting algorithms to learn a set of classifiers [3, 9, 4] or to choose features [7], here we use a linear classifier for object tracking and achieve favorable performance.

To initialize the classifier in the first frame, we draw positive and negative samples around the labeled target location. Suppose the location of the target object in the first frame is denoted by $\mathbf{l}_1 = (x_1, y_1)$, we use a Gaussian perturbation to draw samples in a circular area which satisfies $\| \mathbf{l}_{pos} - \mathbf{l}_1 \| < \gamma$, and draw negative samples in an annular area specified by $\gamma < \| \mathbf{l}_{neg} - \mathbf{l}_1 \| < \eta$, where γ and η are thresholds defining the circle and annular areas, respectively. The sets, $\mathbf{l}_{pos}$ and $\mathbf{l}_{neg}$, denote the locations of positive and negative candidates, respectively. Without loss of generality, we set the scales of the positive and negative candidates the same as our labeled target object. We then crop the images specified by the set of samples $\mathbf{l}_{pos}$ and $\mathbf{l}_{neg}$ and compute the sparse code of each image patch to form the training data, $\{\mathbf{z}_i, y_i\}_{i=1}^M$, where $\mathbf{z}_i \in \mathbb{R}^{n+2d}$, $y_i \in \{+1, -1\}$, and M is the number of training samples.

With the training data, our linear classifier is learned by minimizing the following loss function

$$J(\mathbf{w}) = \frac{1}{M}\sum_{i=1}^{M} \ell(y_i, \mathbf{w}, \mathbf{z}_i) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{3}$$

where $\mathbf{w}$ is the classifier parameter, $\ell(\cdot)$ is a loss function, and λ controls the strength of the regularization term. We use the logistic regression loss function due to its convexity and differentiable properties:

$$\ell(y, \mathbf{w}, \mathbf{z}) = \log\left(1 + e^{-y\mathbf{w}^\top \mathbf{z}'}\right), \tag{4}$$

where $\mathbf{z}' = [\mathbf{z}^\top, 1]^\top$ is the augmented vector. The corresponding classification score with the learned classifier can be computed by

$$h(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{z}'}}. \tag{5}$$

Once the classifier is initialized, the classification score can be utilized as the similarity measure for tracking. A sample with larger classification score indicates it is more likely to be generated from the target class. The most likely sample is considered as the tracking result for that image frame.

4. Proposed Tracking Algorithm

With the sparse representation and the learned linear classifier, we propose a two-stage tracking algorithm based on Bayesian inference which can alleviate the visual drift problem when updating our dictionary and classifier to account for appearance change of the target and background.

4.1. Object Tracking by Bayesian Inference

We estimate the target states (i.e., motion parameters) sequentially using the the Bayesian inference framework. Given the observations of the target $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \ldots \mathbf{z}_t\}$ up to time t, the target state $\mathbf{x}_t$ can be computed by the maximum a posteriori (MAP) estimation:

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p(\mathbf{x}_t | \mathbf{z}_{1:t}). \tag{6}$$

The posterior probability $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ can be inferred by the Bayesian theorem recursively

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}), \quad (7)$$

where $p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}$. With the particle filter method [11], $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ can be approximated by a finite set of particles.

Within the above formulation, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the dynamic model that describes the temporal correlation of the target states in consecutive frames, and $p(\mathbf{z}_t|\mathbf{x}_t)$ is the observation model or likelihood function $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ which estimates the likelihood of a state given an observation. In our algorithm, we model the motion of a target object between two consecutive frames with affine transformation. Let $\mathbf{x}_t$ be the six-dimensional parameter vector for affine transformation. We model the transformation of each parameter independently by a scalar Gaussian distribution between two consecutive frames. Then the dynamic model can be represented by a Gaussian distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \Sigma)$, where Σ is a diagonal covariance matrix whose elements are the variances of the affine parameters. In our method, the observation model is defined by

$$p(\mathbf{z}|\mathbf{x}) \propto h(\mathbf{z}), \quad (8)$$

where $h(\cdot)$ is the classifier defined in Equation 5.

4.2. Two-Stage Object Tracking

The image appearances of both target and background are likely to change due to numerous factors as discussed above. Thus robust visual tracking entails the need to update the observation model, i.e., dictionary and classifier in this work, when new tracking results become available. Online update of observation models have been shown to be effective in the recent tracking literature [12, 17, 9, 19, 10, 4, 18, 13]. However, a naive method that updates the observation model with all the new tracking results may adversely cause visual drifts. For example, updating the model with noisy observations is likely to degrade the discriminative strength of a classifier in separating targets from the background. Several approaches have been proposed to alleviate this problem [17, 4, 13] with demonstrated success when the constraints or assumptions of these methods are satisfied in the image sequences.

The main challenge for updating the observation model is that it is difficult to determine whether the new tracking result is a good positive example (e.g., without alignment error and excluding occluded image regions) since no ground truth is available. For most tracking scenarios, the only ground truth at our disposal is the labeled target image region in the first frame. All the other image observations obtained online are likely to be different from the ground truth to some degree. To alleviate the visual drift problem when updating our dictionary and classifier, we propose a two-stage tracking method in a way similar to [17]. In each frame when the tracking result is obtained, both the dictionary and classifier are updated. It is carried out by reconstructing the dictionary D_t with the tracking result at time t, and by retraining the classifier (parameterized by $\mathbf{w}_t$) with the same method as used in the first frame. The dictionary D_t and classifier parameters $\mathbf{w}_t$ are used to construct an adaptive observation model for particle filtering. To reduce the risk of visual drift, we also retain the dictionary D_1 and classifier parameters $\mathbf{w}_1$ obtained in the first frame for constructing a static observation model based on the ground truth. We use two steps to obtain the tracking result at time t. In the first stage, we use a particle filter to estimate the initial tracking result using the adaptive observation model. From the estimated tracking result, in the second stage we use a particle filter with the static observation model to determine the final tracking result.

The two-stage tracking algorithm is summarized in Algorithm. 1. The first step can effectively avoid the local minimum problem since the appearance change between two consecutive frames is not expected to be too large. The second step can effectively alleviate the visual drift problem since it ensures the final tracking result should be as similar as the only ground truth obtained from the first labeled frame. With this tracking strategy, no thresholds need to be set in order to determine when to update the observation model which is often required by existing algorithms (e.g., [17, 18]). Therefore, more robust results can be obtained by our algorithm.

5. Experiments

We evaluate the performance of the proposed algorithm using several challenging sequences where most of them are publicly available and some are collected on our own. The challenging factors of these sequences are listed in Table 1.

Table 1. Tracking sequences used in our experiments.

Sequences	Main challenging factors
David [19]	large illumination variation, out-of-plane pose change, partial occlusion
Sylvester [19]	out-of-plane pose change, fast motion, illumination change
car [19]	large illumination change, distraction from other objects
jumping [13]	image blur, fast motion
face [1]	long-duration occlusion
singer [14]	large illumination variation, large scale change
PETS2009 [8]	out-of-plane pose change, heavy occlusion
Avatar1	large scale change, low contrast
Avatar2	heavy occlusion, out-of-plane pose change, illumination change, scale change
surfing	fast motion, large scale change, small target

Algorithm 1 Two-stage Tracking Algorithm.

Input: Image frames $F_1, \ldots, F_T$. The target object is labeled in the first frame.
Output: Target state $\hat{\mathbf{x}}_t^*$ at time t, and the object location shown with a bounding box.
for $t = 1, \ldots, T$ **do**
  if $t = 1$ **then**
    Construct an initial over-complete dictionary D_1, and learn a linear classifier with parameter $\mathbf{w}_1$.
  else
    Stage 1. Perform particle filtering to estimate the target state $\hat{\mathbf{x}}_t$ by using the previous tracking result $\hat{\mathbf{x}}_{t-1}^*$, and the adaptive observation model parameterized by D_{t-1} and $\mathbf{w}_{t-1}$.
    Stage 2. Set $\hat{\mathbf{x}}_{t-1}^* = \hat{\mathbf{x}}_t$. Perform particle filtering again with $\hat{\mathbf{x}}_{t-1}^*$ and the static observation model parameterized by D_1 and $\mathbf{w}_1$ to determine the final tracking result $\hat{\mathbf{x}}_t^*$. Plot the tracking result in the current image.
    Update the adaptive tracker to get D_t and $\mathbf{w}_t$ with the tracking result $\hat{\mathbf{x}}_t^*$.
  end if
end for

We compare the performance of the proposed algorithm with five state-of-the-art tracking works including the Incremental Visual Tracking (IVT), L1 tracking (L1T) [18], Multiple Instance Learning tracking (MIL) [4], Visual Tracking Decomposition (VTD) [14], and P-N learning tracking (TLD) [13] methods. The IVT, L1T and VTD methods are generative methods whereas the others are discriminative trackers. For fair evaluation, we use the codes provided by the authors with the same initialized target locations in these sequences. For the IVT, L1T and VTD methods which also use particle filters to estimate the target state, we choose the same dynamic model and parameters as our method. Each object image is normalized to 32×32 pixels from which overlapping 16×16 patches with a shift of 8 pixels are extracted. The number of particles is set to 600 in all experiments. Some tracking results are shown in the next two sections. The tracking videos, MATLAB code, and data sets can be found at `http://faculty.ucmerced.edu/mhyang`.

5.1. Qualitative Evaluation

In the *David* sequence, the ambient light changes from dark to bright in the first few frames, and the scale as well as pose of the target object also vary significantly. All the algorithms performs reasonably well in tracking the target object although the MILT method does not estimate the scale change well. Some representative tracking results are shown in Figure 1 (a). However, when the target object undergoes out-of-plane pose change, the L1T method drifts away from the ground truth locations gradually. The IVT method also fails in some frames but recovers to track the target in subsequence frames.

In the *Sylvester* sequence, there are frequent variations of pose and illumination. The IVT method gradually drifts away from the target object and the L1T algorithm also loses track of the target object for a number of frames. The other methods perform well in most frames of this sequence. For the *car* sequence, there is significant illumination change when the target object passes underneath the trees and overpass. The tracking results (Figure 1 (c)) show that the MILT method does not perform well after the first illumination change. The VTD method also fails when significant illumination change occurs but performs well in most of the frames. On the other hand, the TLD algorithm fails in the last few frames when another car with similar appearance enters the scene. Nevertheless, the IVT, L1T and proposed algorithms perform well in this sequence.

In the *jumping* sequence, there exists drastic image blur due to fast motion of the target object. Some representative tracking results are shown in Figure 1 (d). The IVT, L1T, MILT, TLD and proposed methods perform well while the VTD method has relative larger tracking errors. In the *face* sequence, there are frequent heavy occlusions. The TLD method has large tracking errors when the target object is heavily occluded whereas the IVT and MILT algorithms gradually drifts away. On the other hand, the L1T, VTD and our methods perform reasonably well in this sequence.

Figure 2 presents more tracking results. In the *singer* sequence, there is drastic illumination and scale change of the target object. The TLD method loses track of the target object for most of the frames, and the VTD algorithm also fails during the illumination change. It is worth noticing that we evaluate on a different singer which is more difficult to track than the one used in [14]. The MILT method has large tracking errors since it does not estimate the scale change of the target well, whereas the IVT, L1T and proposed algorithms are able to locate the target object in this sequence. There are multiple objects similar to the target in the *PETS2009* sequence. As these objects move in different directions and occlude each other in the scenes, this image set poses significant challenges for visual tracking. Nevertheless, our method performs better than the other methods and some results are shown in Figure 2 (b). It can be explained by that the proposed appearance model based on local sparse representation is more discriminative than those used in the other methods.

The contrast between the target object and the background is rather low in the *Avatar1* sequence. In addition, there is a significant scale change as the object moves away in the scenes. Some results are presented in Figure 2 (c). The MILT, VTD and TLD methods gradually lose track of the target object while the other methods perform reason-

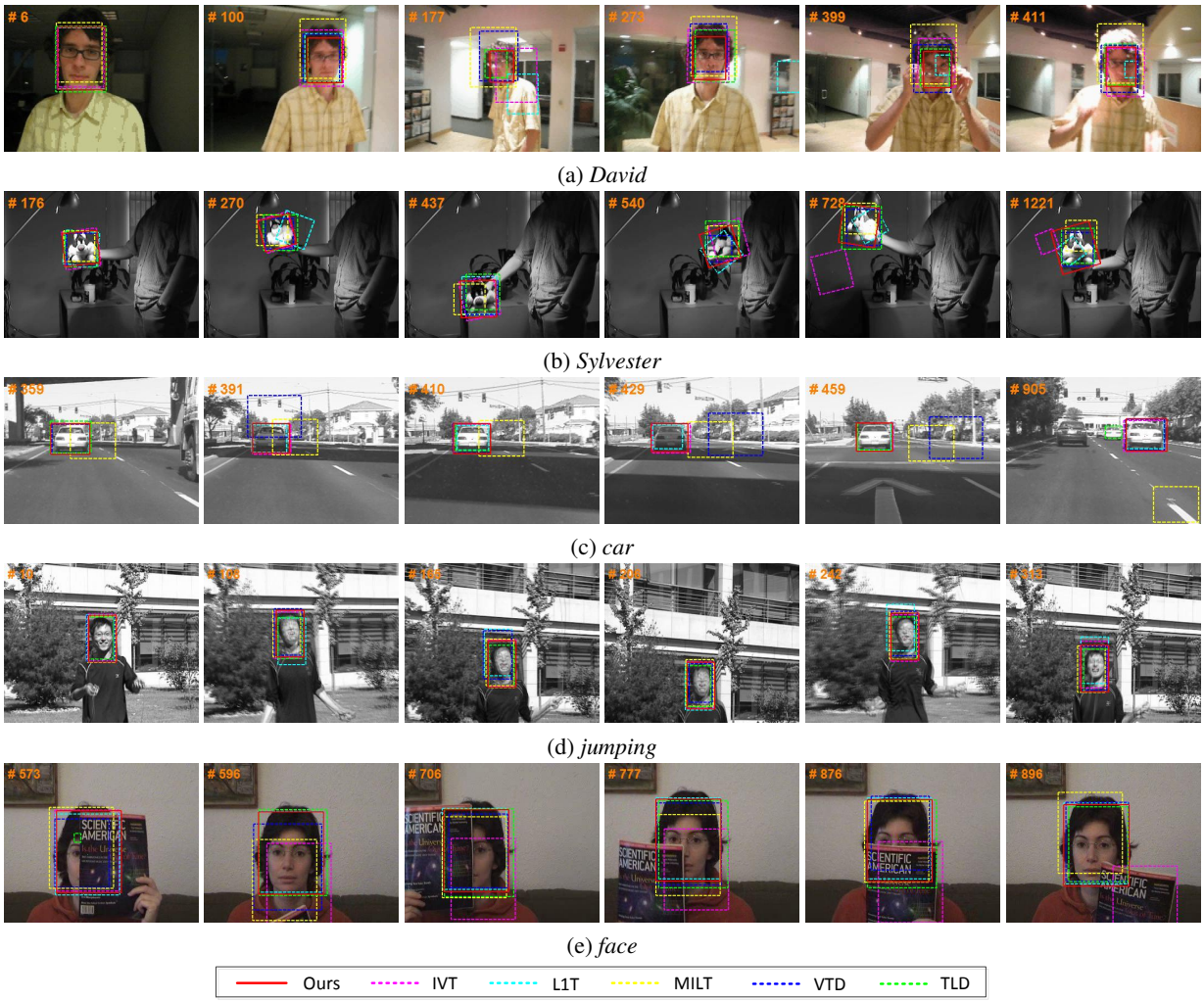

Figure 1. Tracking results on challenging sequences.

ably well in this sequence. In the *Avatar2* video, the target object undergoes scale change, occlusion and out-of-plane pose variation. All the tracking methods succeed in tracking the target object before significant occlusion and out-of-plane pose change occur in frame 141. After that, the contrast between the target and the background becomes much smaller (shown in frame 165) and only our method is able to track the object. In the *surfing* sequence, the target object moves rather fast and there is large scale change. Some sample results are shown in Figure 2 (e). Our method performs well while all the other methods lose track of the target object gradually. It shows that the proposed method with local sparse representation-based object representation and linear classifier model is effective in dealing this challenging sequence.

To demonstrate the effect of our two-stage tracking mechanism, we implement an one-stage method (referred as T1) which uses one single particle filter with the adaptive observation model. All the other components of the T1 tracker are the same as the proposed two-stage method. In addition, we compare the proposed discriminative tracker with local sparse representation against the L1T algorithm (a generative tracker with holistic sparse representation) [18]. We implement an improved L1T method (referred as L1T2) with the proposed two-stage particle filtering method, and all the other components of the L1T2 tracker are the same as the L1T method. We evaluate the T1 and L1T2 methods on the *Avatar2* and *surfing* sequences. Some tracking results are presented in Figure 3 and videos can be found at the web page mentioned above. Both these two trackers lose track of the target objects gradually. The experimental results show that both the two-stage particle filters and discriminative learning as well as local sparse representation are crucial for object tracking in these scenarios.

5.2. Quantitative Evaluation

For quantitative evaluations, we measure the tracking accuracy of all the algorithms on these sequences. We use

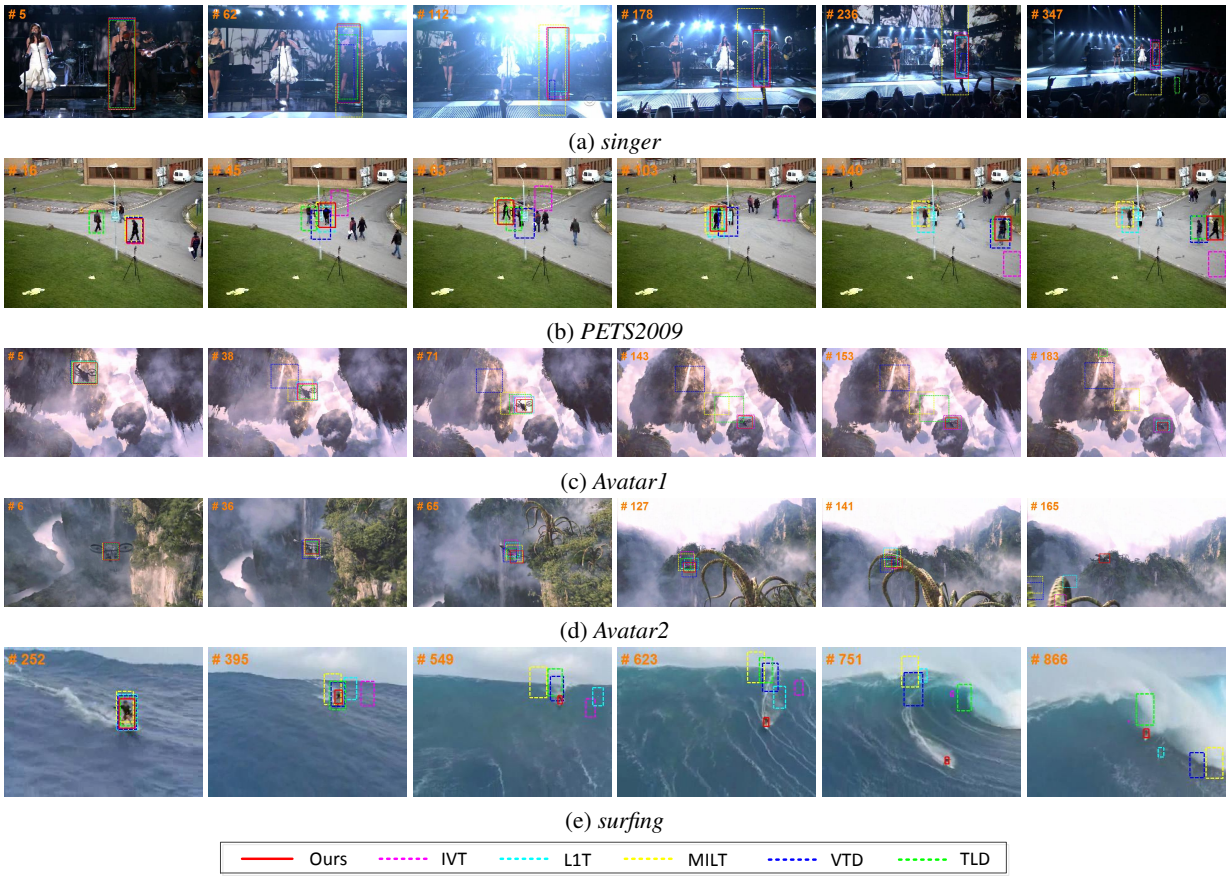

Figure 2. More tracking results on challenging sequences.

the center location error as the criterion for accuracy measure in this work. The center location error is defined as the distance between the central location of the tracked target and the manually labeled ground truth data. The error plots are shown in Figure 4. The quantitative results show that the proposed tracker performs favorably against all the other methods. It is worth noticing that our tracker have lower drifting errors than the others which can be explained by the proposed update mechanism and two-stage tracking algorithm.

5.3. Discussion

From the experimental results with the *singer*, *Avatar1*, *Avatar2* and *surfing* image sequences, it is clear that the proposed method performs well against the other algorithms when the targets undergo large scale changes. The tracking results from the *Sylvester*, *Avatar1* and *Avatar2* sequences demonstrate that our method performs well against the other when the contrasts between the foreground and background are rather low. In addition, the proposed method is able to handle large illumination change as shown in the *David*, *car* and *singer* sequences. Similarly, the proposed method is shown to effectively deal with motion blur in the *jumping* sequence, fast moving objects in the *surfing* sequence, out-of-plane pose variation in the *Sylvester*, *PETS2009* and *Avatar2* sequences. The experimental results from the *face*, *PETS2009* and *Avatar2* sequences demonstrate that our method is able to handle long-duration partial occlusions and short-duration heavy occlusions.

6. Conclusion

In this paper, we propose an online tracking algorithm based on local sparse representation and classifier learning. We use the sparse codes of local image patches with an over-complete dictionary for object representation, and learn a linear classifier to separate the target object from the background. Based on the classification score, we define an observation model and implement object tracking within the Bayesian inference framework. To adapt our tracker to account for appearance change of the target and the background and to alleviate the drift problem when updating our tracker, we propose a two-stage tracking method. Experiments on several challenging sequences with comparisons to state-of-the-art adaptive tracking methods demonstrate the effectiveness of our tracking algorithm. Our future work will focus on large scale experiments to evaluate the state-of-the-art tracking algorithms with benchmark data sets and sound metrics.

Figure 3. Tracking results of the T1 and L1T2 methods on the *Avatar2* and *surfing* sequences.

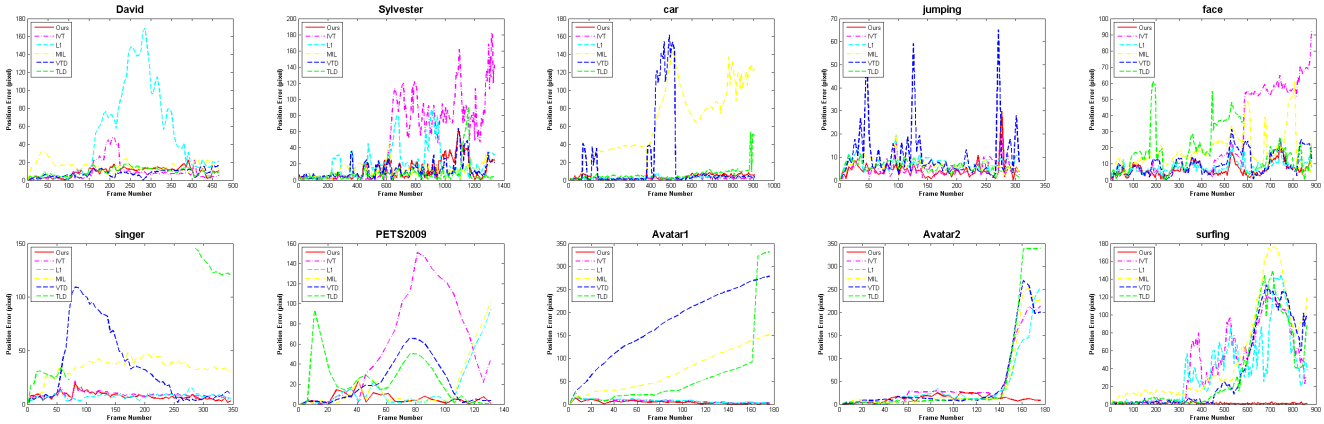

Figure 4. Error plots of all the test sequences.

Acknowledgements

This work was carried out when Q. Wang was a visiting scholar at UC Merced with support in part by the NSFC grant 6107131. M.-H. Yang is supported in part by a faculty start-up fund and a Google Faculty Award.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[2] S. Avidan. Support vector tracking. In *CVPR*, pages 184–191, 2001.

[3] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[5] M. J. Black. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV*, pages 329–342, 1996.

[6] M. J. Black, D. J. Fleet, and Y. Yacoob. A framework for modeling appearance change in image sequences. In *ICCV*, pages 660–667, 1998.

[7] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *PAMI*, 27(10):1631–1643, 2005.

[8] J. Ferryman, J. Crowley, and A. Shahrokni. PETS: Dataset and challenge. In *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[9] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, pages 260–267, 2006.

[10] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247, 2008.

[11] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[12] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[13] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56, 2010.

[14] J. Kwon and K. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

[15] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

[16] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, pages 624–637, 2010.

[17] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26(6):810–815, 2004.

[18] X. Mei and H. Ling. Robust visual tracking using l1 minimization. In *ICCV*, 2009.

[19] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[20] 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.

[21] T.-T. Wu and K. Lange. Coordinate descent algorithms for LASSO penalized regression. *Ann. Appl. Stat.*, 2(1):224–244, 2008.

[22] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.