Online Discriminative Dictionary Learning for Visual Tracking

Fan Yang[†], Zhuolin Jiang[§] and Larry S. Davis[†]
[†]University of Maryland College Park, MD, USA
[§]Noah's Ark Lab, Huawei Technologies
{fyang,lsd}@umiacs.umd.edu, zhuolin.jiang@huawei.com

Abstract

Dictionary learning has been applied to various computer vision problems, such as image restoration, object classification and face recognition. In this work, we propose a tracking framework based on sparse representation and online discriminative dictionary learning. By associating dictionary items with label information, the learned dictionary is both reconstructive and discriminative, which better distinguishes target objects from the background. During tracking, the best target candidate is selected by a joint decision measure. Reliable tracking results and augmented training samples are accumulated into two sets to update the dictionary. Both online dictionary learning and the proposed joint decision measure are important for the final tracking performance. Experiments show that our approach outperforms several recently proposed trackers.

1. Introduction

Although visual tracking has been widely investigated for many years, it is still challenging to perform robust tracking under complex scenarios, such as pose variance, occlusions and cluttered backgrounds. Various algorithms have been proposed to deal with different scenarios in visual tracking with the focus on appearance modeling and decision strategy design [10, 1, 24, 3, 14, 15, 9, 30, 8]. Recently, an increasing number of studies apply sparse coding to visual tracking and generate state-of-the-art results [20, 16, 38, 12, 37, 36, 35].

With superior representative ability, sparse representations can capture the most essential information from a training set and are very robust to noise, which are desirable for appearance modeling in visual tracking since it is not feasible to maintain an arbitrarily large training set explicitly. However, sparse representation based approaches have some drawbacks. First, previous methods either leave the dictionary unchanged during tracking [16] or update it by simply using new samples as dictionary items [12, 38]. Dictionary update is crucial for dealing with changes in appearance, pose and brightness. However, methods using static dictionaries or heuristic dictionary update are unlikely to construct dynamic dictionaries which characterize changing objects well. Additionally, many sparse coding based trackers [20, 12, 37, 36, 35] seek to minimize the reconstruction error to increase the representative power of the learned dictionary. However, considering visual tracking as a binary classification problem, a dictionary learned by minimizing reconstruction error might not have sufficient discriminative power to differentiate the foreground from the background.

Motivated by previous works, we attempt to exploit the discriminative ability of sparse representations for better appearance modeling. We present an online discriminative dictionary learning (ODDL) algorithm for visual tracking which enforces both the reconstructive and discriminative capacity of the dictionary. Apart from the reconstruction error, a specific class label is associated with each dictionary item to enforce discriminability during dictionary learning. In this way, the ODDL algorithm learns a sparse dictionary and a linear classifier simultaneously. The quality of each tracking candidate is measured based on a linear combination of a quadratic appearance distance and a classification error instead of relying on only one of them. To account for target appearance changes, the ODDL algorithm adaptively updates dictionary items and the classifier given new samples in a principled way.

Our contributions are three-folds. First, the ODDL algorithm focuses on both the discriminative and reconstructive power of the dictionary in appearance modeling. The dictionary learning is performed in a joint manner, where the discriminative and reconstructive power are enforced in a unified algorithm. The learned dictionary is able to represent the object well and differentiate the object from the background simultaneously. Second, we propose a joint decision measure to evaluate the reliability of candidates to improve tracking accuracy, in contrast to previous work which only relies on reconstruction error. Finally, the dictionary learned by our approach captures changes to the object's appearance through online updating with a set of adaptively selected, reliable samples. To further accelerate the update, we utilize a batch online learning technique which reduces the computational complexity in optimization. To the best of our knowledge, this is the first work attempting to incorporate both discriminability of a dictionary and efficient online dictionary learning into a unified framework for visual tracking.

2. Related work

There is a rich literature on visual tracking. Several classic algorithms have been proposed and demonstrated impressive performance. In [24], an incremental visual tracker (IVT) using holistic features is presented, but it is less effective in handling high levels of occlusion or non-rigid distortion. The Fragment-based tracker [1] utilizes local patches to address partial occlusion; tracking is done by combining votes of matching local patches using a static template. In [3], an algorithm extends multiple instance learning to an online setting for object tracking, while [25] extends the tracking-by-detection framework with multiple modules for reducing drifts. The visual tracking decomposition (VTD) approach [15] fuses multiple motion and observation models to account for appearance variation without the discriminative ability to separate foreground from background.

Due to the strong representative capcity of sparse coding, many sparse representations have been applied to visual tracking and achieved impressive results. [20] was the first to apply sparse representations to visual tracking. However, it simply uses holistic object samples as templates for the dictionary, without consideration of the background information and computes sparse codes by ℓ_1 minimization. No dictionary learning algorithms and systematic update strategies are adopted, which makes the tracker sensitive to object changes. [16] constructs a dictionary using a K-selection approach before tracking starts. Although it considers background information during dictionary construction, the dictionary is fixed during the entire tracking procedure, thus might not be adaptive to new samples. To better improve the discriminative power, [38] combines a sparsity-based discriminative classifier with a generative model based on both holistic and local representations, where spatial information is also encoded. Nevertheless, the two parts are independent and combined in a heuristic way. [12] proposes an alignment pooling approach to obtain global sparse representations from local patches. The templates are updated to capture object changes by replacing old templates by new ones, but no dictionary learning is adopted. [37] applies the multi-task learning framework using the group sparsity constraints among candidates, where each candidate can be considered as one task. Similarly, [36] extends the approach in [37] by imposing low-rank constraints on the joint optimization of the candidate groups. However, both focus more on candidate selection than good appearance modeling using sparse representations. [35] also represents candidates by the target and background templates to improve the tracker's discriminative ability. Without learning technique, arbitrary selected templates may not account for object appearance changes well. [29] is closely related to our work in that it also incorporates the discriminative power into standard sparse representations by learning a classifier. Nevertheless, the dictionary and classifier are learned separately rather than jointly; additionally, the two-stage tracking approach significantly increases the complexity and makes the tracker unsuitable for any online applications. Our work also differs from some other recent trackers based on sparse representations, which do not learn a dictionary [4], do not use joint decision to update the dictionary [31], or apply non-negativity constraint to the objective function to learn sparse codes [28].

To improve the representative and discriminative power of dictionaries, many dictionary learning approaches have been proposed recently. Unsupervised dictionary learning algorithms aim to minimize the residual for image reconstruction. Specifically, group features with k-means clustering are used in [27]. The K-SVD algorithm [2] generalizes k-means clustering to learns an over-complete dictionary; semantic relationships between dictionary items are also included in [11]. Dictionaries learned by these algorithms reconstruct the objects well but may not be suitable for classification tasks. Recently, supervised dictionary learning has been introduced for better classification. A simple method is to learn a dictionary for each class label, and a test sample is then classified using the label which generates the smallest reconstruction error [21, 19, 18, 26]. Class-specific dictionaries [23]and multiple category-specific dictionaries with a shared common dictionary [39] have also been developed. In [19, 18, 22, 5, 34], discriminative terms are included in the objective function. A structured dictionary with class labels via Fisher discriminative criterion is learned in [33]. However, we note that none of the above dictionary learning algorithms have been applied to visual tracking efficiently and effectively. In this work, we incorporate the discriminative dictionary learning into a tracking framework and tackle the problems of insufficient training samples and efficient online update.

3. Online discriminative dictionary learning

3.1. Problem formulation

Given a set of training samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^{d\times n}$ with class labels $\mathbf{Y} = \{-1, 1\}$, our goal is to learn a compact dictionary which is discriminative to distinguish the object from the background. Each $\mathbf{x}_i$ is a feature vector extracted from an image region corresponding to a positive sample (target object) or a negative sample (background).

Given a dictionary $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_k\} \in \mathbb{R}^{d\times k}$ with k items, $\mathbf{x}_i$ can be reconstructed by a linear combination of a few items from the dictionary, $\mathbf{x}_i \approx \mathbf{D}\mathbf{c}_i$, where $\mathbf{c}_i \in \mathbb{R}^k$ is the sparse code of $\mathbf{x}_i$ and can be computed by:

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}\|^2 + \lambda\|\mathbf{c}\|_1 \tag{1}$$

where λ is a parameter to balance sparsity and the reconstruction error. To learn a discriminative dictionary, $\mathbf{c}_i$ can be used as a feature descriptor and incorporated into a supervised learning framework:

$$\min_{\mathbf{D},\mathbf{W}} \sum_i \ell(y_i, f(\mathbf{c}_i, \mathbf{W})) + \lambda \|\mathbf{W}\|_F^2 \tag{2}$$

where ℓ is the loss function and f is a classifier with classification parameters $\mathbf{W} \in \mathbb{R}^{m\times k}$. Motivated by [13], we assign a specific label to each dictionary item in Equation 2. We hope that the samples from class m will typically be represented by the dictionary items from class m. In addition, in order to make the learned $\mathbf{D}$ good for classification, we learn the classifier and the dictionary simultaneously. Hence we incorporate an ideal sparse coding error and a linear regression loss into the objective function of dictionary learning

$$\begin{aligned} \min_{\mathbf{D},\mathbf{W}} \quad & \sum_i \ell(\mathbf{D}, \mathbf{W}; \mathbf{x}_i, \mathbf{f}_i, \mathbf{l}_i) + \lambda \|\mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{c}_i = \arg\min_{\mathbf{c}} \|\mathbf{x}_i - \mathbf{D}\mathbf{c}\| + \gamma\|\mathbf{c}\|_1, i = 1, ..., n \end{aligned} \tag{3}$$

where $\ell(\mathbf{D}, \mathbf{W}; \mathbf{x}_i, \mathbf{f}_i, \mathbf{l}_i) = (1-\mu)\|\mathbf{f}_i - \mathbf{W}\mathbf{c}_i\|_2^2 + \mu\|\mathbf{l}_i - \mathbf{c}_i\|_2^2$ is the loss function given a new sample $\mathbf{x}_i$, $\mathbf{f}_i$ and $\mathbf{l}_i$. $\|\mathbf{x}_i - \mathbf{D}\mathbf{c}_i\|^2$ is the reconstruction error. $\|\mathbf{l}_i - \mathbf{c}_i\|^2$ is the ideal sparse code error, where $\mathbf{l}_i = [l_{i,1}, l_{i,2}, ..., l_{i,k}]^T = [1, ..., 0, 1, ..., 0]^T \in \mathbb{R}^k$ is an ideal sparse code for $\mathbf{x}_i$. If $l_{i,k} = 1$, the training samples $\mathbf{x}_i$ and dictionary item $\mathbf{d}_k$ share the same label, while $l_{i,k} = 0$ means they belong to different classes. $\|\mathbf{f}_i - \mathbf{W}\mathbf{c}_i\|^2$ is the quadratic loss for linear regression. $\mathbf{f}_i = [0, ..., 1, ..., 0]^T \in \mathbb{R}^m$ is the label vector for $\mathbf{x}_i$ where the non-zero position indicates the class label of $\mathbf{x}_i$. m is set to 2 for the tracking problem. The parameter μ controls the contributions of the ideal sparse code error and the linear regression error.

The dictionary $\mathbf{D}$ learned from Equation 3 is both reconstructive and discriminative since we impose the label information for dictionary items and classification error during the optimization.

3.2. Optimization

This objective function in Equation 3 is nonlinear and nonconvex, so we resort to stochastic gradient descent. The gradient with respect to $\mathbf{W}$ is

$$\frac{\partial\ell}{\partial\mathbf{W}} = (1-\mu)(\mathbf{W}\mathbf{c}_i - \mathbf{f}_i)\mathbf{c}_i^T + \lambda\mathbf{W} \tag{4}$$

However, the dictionary $\mathbf{D}$ is not explicitly defined in ℓ but implicitly defined on the sparse code $\mathbf{c}_i$. To obtain the gradient with respect to $\mathbf{D}$, we adopt the implicit differentiation algorithm on the fixed point equations as in [17, 32]. By applying the chain rule we obtain

$$\frac{\partial\ell}{\partial\mathbf{D}} = \frac{\partial\ell}{\partial\mathbf{c}_i}\frac{\partial\mathbf{c}_i}{\partial\mathbf{D}} \tag{5}$$

where $\frac{\partial\ell}{\partial\mathbf{c}_i} = (1-\mu)\mathbf{W}^T(\mathbf{W}\mathbf{c}_i - \mathbf{f}_i) + \mu(\mathbf{c}_i - \mathbf{l}_i)$. To calculate $\frac{\partial\mathbf{c}_i}{\partial\mathbf{D}}$, we define the fixed point equation $\mathbf{D}^T(\mathbf{D}\mathbf{c} - \mathbf{x}) = -\lambda sign(\mathbf{c})$ where the sign function is applied to the elements of $\mathbf{c}$ individually. Then we obtain the derivative of $\mathbf{D}$ by $\frac{\partial\mathbf{c}_\Delta}{\partial\mathbf{D}_\Delta} = (\mathbf{D}_\Delta^T\mathbf{D}_\Delta)^{-1}(\frac{\partial\mathbf{D}_\Delta^T\mathbf{x}}{\partial\mathbf{D}_\Delta} - \frac{\partial\mathbf{D}_\Delta^T\mathbf{D}_\Delta}{\partial\mathbf{D}_\Delta}\mathbf{c})$, where Δ indicates the indices of all non-zero values in $\mathbf{c}$ and $\bar{\Delta}$ indicates the indices of all zeros values. We define an auxiliary variable $\varphi \in \mathbb{R}^k$ where $\varphi_{\bar{\Delta}} = \mathbf{0}$ and $\varphi_\Delta = \frac{\partial\ell}{\partial\mathbf{c}_i}(\mathbf{D}_\Delta^T\mathbf{D}_\Delta)^{-1}$. Therefore, Equation 5 is calculated as

$$\frac{\partial\ell}{\partial\mathbf{D}} = -\mathbf{D}\varphi\mathbf{c}_i^T + (\mathbf{x}_i - \mathbf{D}\mathbf{c}_i)\varphi^T \tag{6}$$

In this way, the gradients with respect to $\mathbf{W}$ and $\mathbf{D}$ are available. We use the learning rate used in [17] which is set to $\min(\eta, \eta i_0/i)$ where η is a constant, $i_0 = M/10$ and M is the iteration number. The online learning procedure is presented in Algorithm 1.

Algorithm 1 Online Discriminative Dictionary Learning

```
Input: Training samples $\mathbf{X} \in \mathbb{R}^{d\times N}$ with labels $\mathbf{Y} \in \mathbb{R}^N$, $\mathbf{W}^0$, $\mathbf{D}^0$, $\mathbf{L}$, $\gamma$, $\lambda$, $M$
Output: New $\mathbf{W}$ and $\mathbf{D}$
for $m = 1$ to $M$
  Permute training samples $\mathbf{X}$
  for $i = 1$ to $N$
    Derive $\mathbf{f}_i$ from $\mathbf{y}_i$;
    Compute sparse code $\mathbf{c}_i$ using Equation 1;
    Find the active set $\Delta_i$ and compute the auxiliary variables $\varphi_i$;
    Set the learning rate $\eta_m = \min(\eta, \eta i_0/i)$;
    Compute the gradient of $\mathbf{W}$ and $\mathbf{D}$ using Equation 4 and 6;
    Update $\mathbf{W}$ and $\mathbf{D}$ by
    $\mathbf{W}^m = \mathbf{W}^m - \eta_m\frac{\partial\ell_i}{\partial\mathbf{D}^m}$, $\mathbf{D}^m = \mathbf{D}^m - \eta_m\frac{\partial\ell_i}{\partial\mathbf{W}^m}$
  end for
  Let $\mathbf{W}^{m+1} = \mathbf{W}^m$ and $\mathbf{D}^{m+1} = \mathbf{D}^m$
end for
```

3.3. Initialization

We run K-SVD [2] on positive and negative samples separately to form two dictionaries with the same size. Then we combine them together to form the initial dictionary $\mathbf{D}^0$. During subsequent learning, the label for each dictionary item remains unchanged since we only update the value of $\mathbf{d}_i$ but keep its label. Given the initial $\mathbf{D}^0$, we compute the sparse code $\mathbf{c}_i$ for $\mathbf{x}_i$ to form the matrix $\mathbf{C}$ containing sparse codes of all samples, and then apply the ridge regression model: $\mathbf{W} = arg \min_{\mathbf{W}} \|\mathbf{F} - \mathbf{WC}\|^2 + \lambda_1 \|\mathbf{W}\|_2^2$ to initialize $\mathbf{W}^0$, where $\mathbf{F}$ is the label matrix for $\mathbf{X}$. The solution to this model is $\mathbf{W} = \mathbf{FC}^T(\mathbf{CC}^T + \lambda_1 \mathbf{I})^{-1}$.

3.4. Classification

Once we have learned the dictionary, we can classify a test sample $\mathbf{x}$. The key idea is combining the similarity between $\mathbf{x}$ and the training samples with the classification score from the classifier. Given a new sample $\mathbf{x}$, we first compute its sparse code $\mathbf{c}$ based on Equation 1. Then, a joint decision measure during testing is defined as

$$\varepsilon(\mathbf{x}) = \|\mathbf{x}_{tr} - \mathbf{Dc}\|^2 + \rho \|\mathbf{f} - \mathbf{Wc}\|^2 \tag{7}$$

where $\mathbf{x}_{tr}$ is the weighted average of the elements in a set (see Sec. 4), $\|\mathbf{x}_{tr} - \mathbf{Dc}\|^2$ is the quadratic appearance distance between the reconstructed sample $\mathbf{Dc}$ and $\mathbf{x}_{tr}$, $\|\mathbf{f} - \mathbf{Wc}\|^2$ is the linear regression loss and ρ is a constant to control the contribution of the linear regression loss. $\mathbf{f} = [1, 0]^T$ is a label indicator which determines a perfect positive sample. By using the joint decision measure, we have a more reliable decision score for the sample $\mathbf{x}$.

4. Tracking procedure

In the first frame, the target is annotated with a bounding box $x^1 = (c_x^1, c_y^1, s^1)$, where (c_x^1, c_y^1) is the centroid position and s^1 is its scale. The superscripts of variables denote frame indices. We randomly and repeatedly shift the bounding box by a few pixels around x^1 to obtain positive samples $\mathbf{X}_+$, and then shift it far away from x^1 to obtain negative samples $\mathbf{X}_-$ without overlap with $\mathbf{X}_+$, to obtain $\mathbf{D}^0$ and $\mathbf{W}^0$ as described in Section 3.3. Tracking is done by inferring the current state x^t of the target from previous states. x^t is selected from P candidates which are randomly sampled around the previous state x^{t-1} from a Gaussian distribution $p(x^t|x^{t-1})$. We choose the candidate with the smallest joint decision error ε using Equation 7 as the tracking result.

To compute the reconstruction error $\varepsilon_{rec} = \|\mathbf{x}_{tr} - \mathbf{Dc}\|^2$ in Equation 7, we accumulate the feature extracted from the bounding box at the optimal location into a set T. The optimal location is determined by the tracking result using our deterministic tracker. Optimal locations from current frames are added to T while those from older frames are deleted from T, so that T has a fixed number of elements, denoted as U_1. We associate with each element in T the weight $w = e^{-\varepsilon}$, where ε is the joint decision error. $\mathbf{x}_{tr}$ in Equation 7 is then computed as the weighted average of the elements in T since elements in T with different reliability should have different importance on the combined sample $\mathbf{x}_{tr}$. Initially, T contains just one element - the bounding box used to initialize the tracker. Its weight is 1.

To update $\mathbf{D}$ and $\mathbf{W}$ periodically, we construct another set, S. In each frame, after determining the optimal location of the bounding box, we randomly sample bounding boxes around the optimal location as positive samples, and far away from the optimal location as negative samples. By controlling the distance from the optimal location, we ensure that most negative samples contain pure background so that they differentiate from the target to the most extent. These samples are added to S. When S reaches a critical size U_2, we apply the ODDL algorithm to update the dictionary, and then empty S.

When accumulating elements into T and S, the tracking result may contain significant noise and thus is not reliable if the optimal location of the bounding box determined by our tracker has a high reconstruction error $\varepsilon_{rec} = \|\mathbf{x}_{tr} - \mathbf{Dc}\|^2$ or a high classification error $\varepsilon_{cls} = \|\mathbf{f} - \mathbf{Wc}\|^2$. In this case, we skip this frame to avoid introducing noise into T and S. A visualization of the construction of sets T and S is presented in Figure 1. The entire tracking procedure is summarized in Algorithm 2.

5. Experiments

5.1. Experiment Setting

We implemented our tracker in Matlab without code optimization. Since we do not update the dictionary every frame, our implementation is very efficient. The average fps is 5; online dictionary learning takes a few seconds on an i7 3.4GHz desktop with 12G memory. The parameter settings are as follows. To avoid inefficient pooling from local patches, the feature representation for the object in our work is a 496-dimensional histograms of oriented gradients (HoG) feature [6] as it performs better than color features. The dictionary size is fixed to 200 which contains 100 items for positive samples and 100 items for negative samples. More items lead to higher accuracy but slow down the tracker during tracking. Experiments show that using 200 items achieves a good trade-off between accuracy and efficiency. The iteration numbers for initialization and online learning are 5 and 30, respectively. Online learning rate is 0.2. In the first frame, both the numbers of positive and negative samples are 200 and both the numbers for update are 100. For tracking, $U_1 = 20$ and $U_2 = 5$ are for T and S. The candidate number of random samples is 800 in each frame. These parameters are empirically determined and fixed for all sequences.

We compare our tracker with 8 state-of-the-art trackers on 9 public sequences from [24, 3, 15, 30, 12]. To com-

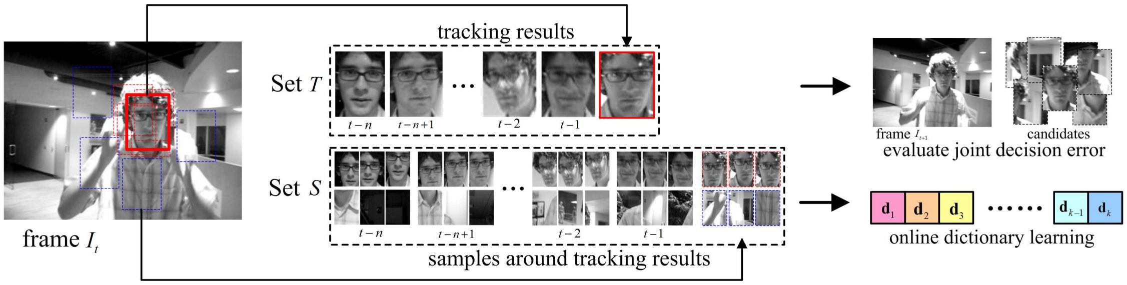

Figure 1. Construction of sets T and S. In frame I_t, the optimal location of the bounding box is added into T for joint decision error evaluation in frame I_{t+1}. Positive samples around the optimal location and negative samples far from the optimal location are added into S for online dictionary learning.

Algorithm 2 Tracking by ODDL

```
Input: Frames $I_1, I_2, ..., I_t$.
Output: Tracking results in each frame $x^1, x^2, ..., x^t$
Initialization $I_t$ ($t = 1$)
    Given initial $x^1 = (c_x^1, c_y^1, s^1)$, sample $N^+$ positive and $N^-$ negative samples;
    Extract features to form $\mathbf{X}$ with label $\mathbf{Y}$;
    Initialize $\mathbf{D}^0$ and $\mathbf{W}^0$;
    Add $\mathbf{X}$ into set $S$, and add the initial state $x^1$ into set $T$;
For each new frame $I_t$ ($t > 1$)
    Sample $P$ candidates around the tracked object $x^{t-1}$ according to distribution $p(x^t|x^{t-1})$ and extract features;
    Compute the sparse code $\mathbf{c}$ for each candidate;
    Apply Equation 7 to each candidate to compute $\varepsilon$ using $\mathbf{D}$, $\mathbf{W}$ and elements in $T$;
    Select the candidate with the smallest $\varepsilon$ as the tracking result $x^t$;
    Sample $N_{new}^+$ positive samples around $x^t$ and $N_{new}^-$ negative samples far away from $x^t$ to obtain new samples $\mathbf{X}_{new}$;
    Add tracking result $x^t$ into $T$ and $\mathbf{X}_{new}$ into $S$;
    If $length(T) > U_1$, remove the oldest element from $T$;
    If $length(S) = U_2$, apply Equation 3 to all elements in $S$ to update $\mathbf{D}$ and $\mathbf{W}$; then empty $S$ for future samples;
    Output $x^t$ and proceed to the next frame $I_{t+1}$.
```

pare the performance of our tracker with other state-of-the-art sparse representation based trackers, we choose the ℓ_1 tracker [20], local sparse appearance tracker (LSK) [16], multi-task tracker (MTT) [37] and two-stage sparse representation tracker (TSP) [29] in our experiments. 4 classic trackers are included, which are the incremental visual tracking (IVT) [24], FragTrack [1], visual tracking decomposition (VTD) tracker [15] and Multiple Instance Learning (MIL) tracker [3]. The test sequences include common scenarios in visual tracking, such as fast motion, pose changes, occlusion, scale change and blurring. Therefore, they can verify the effectiveness of our tracker thoroughly.

Some qualitative tracking results are shown in Figure 2, and quantitative comparisons are summarized in Table 1 and 2. Results of our tracker are averaged from 5 runs on each sequence. For quantitative results, we use the average center location error (CLE) and successful tracking rate (STR). In computing STR, we employ the PASCAL score [7] which is obtained by $s = \frac{area(R_{GT} \cap R_T)}{area(R_{GT} \cup R_T)}$ where R_{GT} and R_T are groundtruth region and tracked result.

5.2. Results

Our tracker is able to handle various scenarios in the testing sequences, including fast motion, pose changes, occlusion, scale change and blurring.

In the *animal* sequence, the target object undergoes large motion and some frames are blurred. Most tracking methods fail (IVT and Frag) or drift far away from the groundtruth (ℓ_1 and LSK). The MIL, MTT and our method produce comparable results, but our method has better STR, which indicates more stable results.

In the *car4* and *singer* sequences, the scale of the target objects greatly changes and there are large illumination variations. Our tracker produces the best and second best STR, respectively, though its CLE does not outperform that of the ℓ_1 tracker.

In the *david* sequence where the person displays a variety of poses, our method outperforms MIL, VTD, Frag and MTT method. Additionally, the STR of our method is far better than these methods, and comparable to the results of the IVT, LSK and ℓ_1 trackers. In the sequence *bolt*, the running athlete Bolt exhibits significant pose changes. Our tracker achieves the best results in terms of CLE and STR, both of which are far better than the compared trackers. The ℓ_1 tracker generates similar CLE to our tracker. But its inability to adjust the size of the tracking window properly leads to low STR.

In the *football* sequence, the athlete runs across the foot-

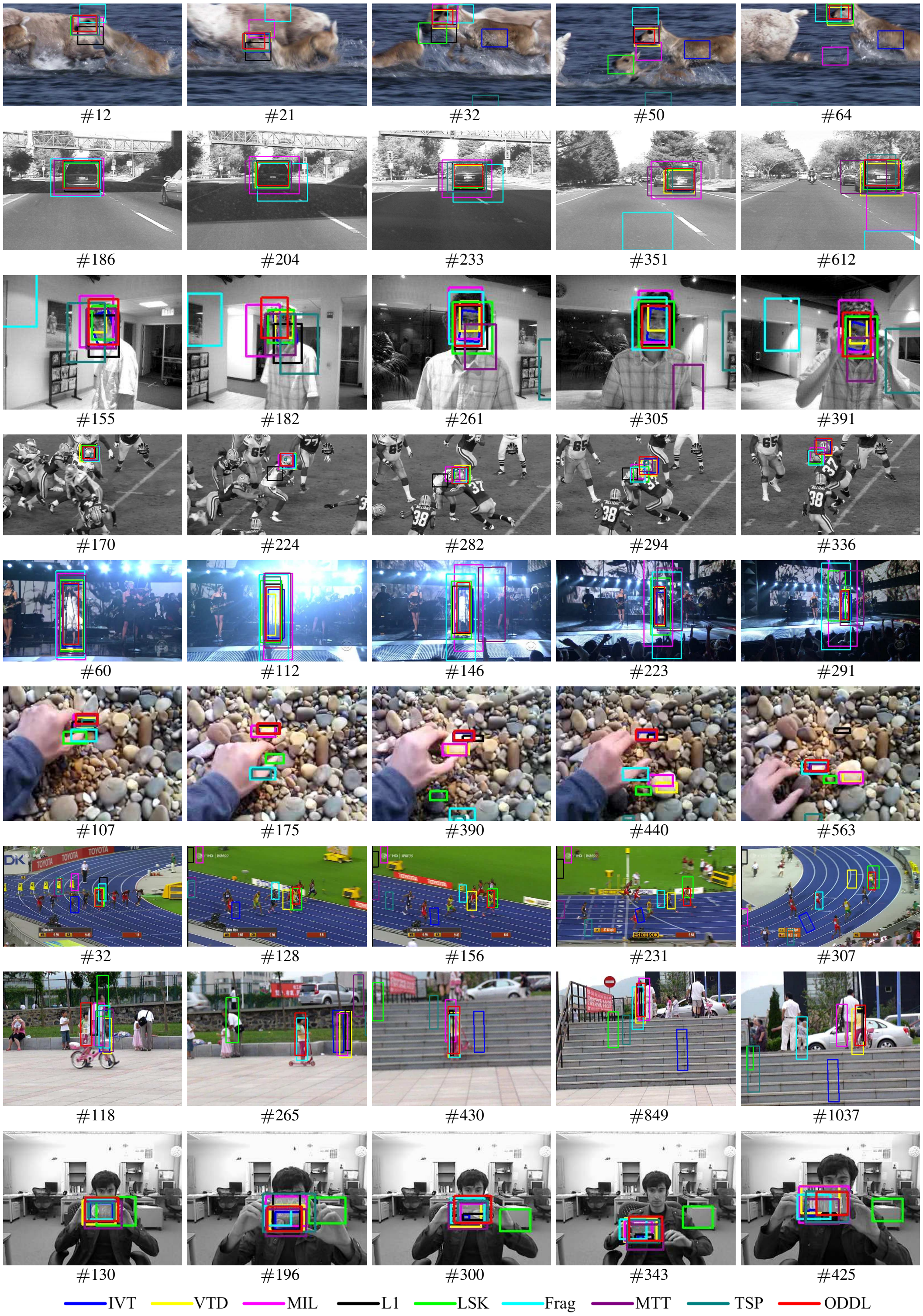

Figure 2. Tracking results using our tracker and state-of-the-art trackers on sequences *animal*, *car4*, *david*, *football*, *singer*, *stone*, *bolt*, *girl* and *twinnings*.

ball field with large pose variance; there is also partial occlusion at the end of the sequence. The IVT, MIL and our method produce comparable results in terms of CLE and STR. When occlusion occurs or the athlete undergoes large motion, the ℓ_1, Frag, LSK and VTD methods are not able to locate the target object accurately. Moreover, the

STR's of these methods are much lower than IVT, MIL and our method. The *stone* sequence is challenging because there are many stones in the background which share similar color and shape with the target object. During the sequence, the target object is also occluded by another stone with similar appearance, which adds more difficulty to successful tracking. Our method successfully keeps track of the target object with the best STR, though the IVT and MTT trackers produce lower CLE.

The sequence *girl* shows a more complex scenario. Despite the full occlusion, blurring and scale changes, our tracker is able to keep track of the object with the best CLE and STR. All the other trackers fails when the occlusion occurs. On sequence *twinnings*, our tracker achieves the second best result in terms of STR.

5.3. Discussion

We evaluate the reliability of our joint decision measure to demonstrate the importance of combing the quadratic distance (between testing sample and T) and linear regression loss. Additionally, we also deactivate online dictionary learning or adopt blind update every frame to see how the online dictionary learning affects tracking performance. The quantitative comparisons are summarized in Table 1 and Table 2.

If we use only the reconstruction error ε_{rec} or classification error ε_{cls}, the results are worse than the method using the joint decision measure both in terms of CLE and STR on all sequences. Therefore, the joint decision measure strategy indeed improves the tracking accuracy. Moreover, removing the online update deteriorates tracking performance, resulting in larger CLE on 6 sequences, and lower STR on 7 sequences. On the other hand, if we blindly update the dictionary every frame without considering the confidence of ε_{rec} and ε_{cls}, the tracker is prone to learn the appearance of the background, leading to higher CLE on 6 sequences and lower STR on 7 sequences. Note that in sequences *car4*, *david* and *football*, the tracker with blind update outperforms the tracker with adaptive update strategy in terms of CLE. We conjecture that the reason is that the target objects in these sequences do not change appearance abruptly. Therefore, even we blindly update the dictionary using all tracked results, the dictionary does not learn much from the background.

From the above experimental results, we conclude that the online learning of dictionary and joint decision measure both contribute to the performance of the proposed tracker.

6. Conclusion

We presented a tracking framework based on sparse representation with online discriminative dictionary learning. The learned dictionary is both reconstructive and discriminative, which allows it to better distinguish the target from the background. During tracking, the best candidate is selected by jointly evaluating the quadratic appearance distance and classification error. Reliable tracking results and augmented training samples are accumulated into two sets respectively for ODDL to update its dictionary. Experimental results demonstrate that our method performs favorably against state-of-the-art trackers and is able to handle various scenarios. Both the online dictionary learning and joint decision measure contribute to the good tracking performance of our tracker.

Acknowledgements

This work was sponsored by Air Force Research Laboratory (AFRL) under contract FA875012C0117.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006. 1, 2, 5

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Proc.*, 54(11):4311–4322, 2006. 2, 3

[3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009. 1, 2, 4, 5

[4] M. Barnard, W. Wang, J. Kittler, S. M. Naqvi, and J. A. Chambers. A dictionary learning approach to tracking. In *ICASSP*, pages 981–984, 2012. 2

[5] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, pages 2559–2566, 2010. 2

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005. 4

[7] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, 2010. 5

[8] J. Fan, X. Shen, and Y. Wu. Scribble tracker: A matting-based approach for robust tracking. *PAMI*, 34(8):1633–1644, 2012. 1

[9] J. Fan, Y. Wu, and S. Dai. Discriminative spatial attention for robust tracking. In *ECCV*, pages 480–493, 2010. 1

[10] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, pages 260–267, 2006. 1

[11] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, pages 487–494, 2010. 2

[12] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829, 2012. 1, 2, 4

[13] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*, pages 1697–1704, 2011. 3

[14] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56, 2010. 1

Table 1. Average center location error (CLE). Trackers using the reconstruction error only, using the linear regression loss only, without online update and with blind update are denoted as ε_{rec}, ε_{cls}, ODDL- and ODDL+. Red is the best and blue is the second best.

Sequence	IVT	VTD	MIL	ℓ_1	LSK	Frag	MTT	TSP	ODDL	ε_{rec}	ε_{cls}	ODDL-	ODDL+
animal	127.5	**12.0**	66.5	26.6	70.0	92.1	**9.2**	216.8	14.9	16.3	145.5	16.4	131.0
car4	**2.9**	12.3	60.1	4.1	**3.3**	179.8	37.2	3.6	4.3	16.2	54.8	9.4	6.2
david	**3.6**	13.6	16.2	7.7	**4.9**	76.7	36.5	73.4	7.5	10.8	59.3	18.4	8.2
football	6.3	**4.0**	13.7	29.9	14.6	16.3	8.0	14.6	6.3	9.2	40.2	6.3	**4.2**
singer	8.6	**4.1**	15.2	**3.2**	14.6	22.1	41.3	6.3	9.3	16.1	35.3	9.4	43.9
stone	**2.3**	31.4	32.3	19.2	68.7	65.9	**2.5**	97.8	4.8	10.3	17.1	4.7	33.2
bolt	189.2	38.6	376.5	377.8	10.3	96.8	386.1	317.0	**9.5**	371.2	**9.8**	117.0	372.1
girl	154.8	50.6	53.1	50.2	244.9	67.2	576.6	201.2	**14.7**	267.2	73.0	44.9	**37.8**
twinnings	14.1	8.6	**6.4**	10.9	61.9	11.3	11.6	**4.1**	8.2	20.4	27.3	41.0	18.0

Table 2. Successful tracking rate (STR). Trackers using the reconstruction error only, using the linear regression loss only, without online update and with blind update are denoted as ε_{rec}, ε_{cls}, ODDL- and ODDL+. Red is the best and blue is the second best.

Sequence	IVT	VTD	MIL	ℓ_1	LSK	Frag	MTT	TSP	ODDL	ε_{rec}	ε_{cls}	ODDL-	ODDL+
animal	0.28	0.78	0.14	0.58	0.41	0.02	0.79	0.31	**0.86**	0.61	0.21	**0.82**	0.32
car4	**0.99**	**0.99**	0.27	**0.99**	**0.98**	0.27	0.38	**0.99**	**0.99**	0.95	0.44	**0.99**	**0.99**
david	0.91	0.37	0.42	0.86	**0.98**	0.16	0.40	0.29	**0.92**	0.71	0.03	0.45	0.73
football	**1.00**	0.98	0.69	0.52	0.77	0.67	0.86	0.32	0.98	0.94	0.18	**0.99**	**0.99**
singer	0.94	0.95	0.23	**0.98**	0.49	0.23	0.34	**0.98**	**0.95**	0.68	0.22	**0.95**	0.45
stone	0.67	0.60	0.50	0.29	0.10	0.24	**0.84**	0.09	**0.87**	0.59	0.20	0.77	0.52
bolt	0.02	0.35	0.04	0.06	0.43	0.10	0.02	0.01	**0.68**	0.02	**0.64**	0.31	0.02
girl	0.09	**0.61**	0.36	0.09	0.09	0.53	0.09	0.09	**0.76**	0.06	0.46	0.34	**0.61**
twinnings	0.28	**0.85**	0.53	0.34	0.23	0.39	0.57	**0.80**	**0.80**	0.65	0.44	0.40	0.38

[15] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010. 1, 2, 4, 5

[16] B. Liu, J. Huang, L. Yang, and C. A. Kulikowski. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, pages 1313–1320, 2011. 1, 2, 5

[17] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *PAMI*, 34(4):791–804, 2012. 3

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

[19] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *ECCV (3)*, pages 43–56, 2008. 2

[20] X. Mei and H. Ling. Robust visual tracking using L1 minimization. In *ICCV*, pages 1436–1443, 2009. 1, 2, 5

[21] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *PAMI*, 30(7):1243–1256, 2008. 2

[22] D.-S. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *CVPR*, 2008. 2

[23] I. Ramírez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, pages 3501–3508, 2010. 2

[24] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008. 1, 2, 4, 5

[25] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *CVPR*, pages 723–730, 2010. 2

[26] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos. Positive definite dictionary learning for region covariances. In *ICCV*, pages 1013–1019, 2011. 2

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010. 2

[28] N. Wang, J. Wang, and D.-Y. Yeung. Online robust nonnegative dictionary learning for visual tracking. 2013. 2

[29] Q. Wang, F. Chen, W. Xu, and M.-H. Yang. Online discriminative object tracking with local sparse representation. In *WACV*, pages 425–432, 2012. 2, 5

[30] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, pages 1323–1330, 2011. 1, 4

[31] Y. Xie, W. Zhang, C. Li, S. Lin, Y. Qu, and Y. Zhang. Discriminative object tracking via sparse representation and online dictionary learning. *IEEE Transactions on Cybernetics*, 2013. 2

[32] J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009. 3

[33] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, pages 543–550, 2011. 2

[34] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, pages 2691–2698, 2010. 2

[35] S. Zhang, H. Yao, H. Zhou, X. Sun, and S. Liu. Robust visual tracking based on online learning sparse representation. *Neurocomputing*, pages 31–40, 2013. 1, 2

[36] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In *ECCV (6)*, pages 470–484, 2012. 1, 2

[37] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, pages 2042–2049, 2012. 1, 2, 5

[38] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pages 1838–1845, 2012. 1, 2

[39] N. Zhou, Y. Shen, J. Peng, and J. Fan. Learning inter-related visual dictionary for object recognition. In *CVPR*, pages 3490–3497, 2012. 2