Discriminative Low-Rank Tracking

Yao Sui, Yafei Tang, Li Zhang
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
suiyao@gmail.com, tangyf24@chinaunicom.cn, chinazhangli@tsinghua.edu.cn

Abstract

Good tracking performance is in general attributed to accurate representation over previously obtained targets or reliable discrimination between the target and the surrounding background. In this work, we exploit the advantages of the both approaches to achieve a robust tracker. We construct a subspace to represent the target and the neighboring background, and simultaneously propagate their class labels via the learned subspace. Moreover, we propose a novel criterion to identify the target from numerous target candidates on each frame, which takes into account both discrimination reliability and representation accuracy. In addition, with the proposed criterion, the ambiguity in the class labels of the neighboring background samples, which often influences the reliability of discriminative tracking model, is effectively alleviated, while the training set is still kept small. Extensive experiments demonstrate that our tracker performs favourably against many other state-of-the-art trackers.

1. Introduction

Visual tracking plays an important role in computer vision for its various practical applications, *e.g.*, video surveillance, human-machine interface and robotics. The fundamental task of visual tracking is to estimate the motion states of the target on each frame, given an initial state. In general, tracking models can be viewed as either generative or discriminative. Generative model focuses on finding a region of interest (ROI) in the frame image as the target, which best matches a learned target appearance model, while discriminative model trains a binary classifier to distinguish the target from background. Current studies have shown that discriminative model achieves better performance if the size of training set is sufficiently large [15], and that generative model can obtain higher generalization when only a limited number of training samples are available [22]. In the pursuit of precise tracking results, we may prefer discriminative model if we can acquire a sufficiently large number of training samples. However, it is unpractical to obtain so many training samples. Even we can acquire these samples, considering the time-sensitive nature of visual tracking, the training cost over these samples is also unacceptably expensive.

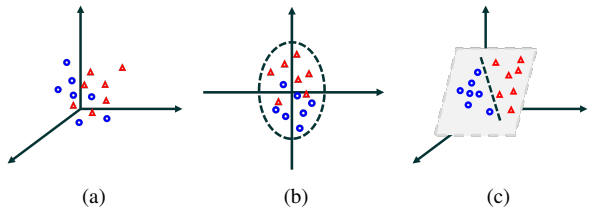


Figure 1. Illustration of the discriminative low-rank learning. (a) A number of original samples belonging to two classes in the three-dimensional observation space. (b) The samples projected into the learned two-dimensional subspace. (c) The samples reconstructed over the learned subspace, which are also successfully separated between the two classes by the learned classifier.

In this work, we aim to utilize the good generalization capability of generative model, and augment it with discriminative capability, in order to achieve improved tracking performance. Thus, our tracking model is both generative and discriminative. We choose two simple but very effective methods for our model: subspace learning and linear classification. Joint learning [26, 25, 19] is used to construct this model, *i.e.*, we construct a subspace to represent the target and background, and simultaneously propagate their class labels by a linear classifier. This assigns both representation and discrimination capabilities to the learned subspace: the target and background can be accurately represented (reconstructed) by the learned subspace, and simultaneously their reconstructions over this subspace can also be reliably distinguished by the linear classifier. In brief, we intend to construct such a subspace, the *discriminative subspace*, over which the reconstructed samples are linearly separable. The basic idea of our method, called the *discriminative low-rank learning*, is illustrated in Fig. 1. With the choices for our tracking model, we are still facing several problems addressed as follows.

1.1. Representation

Subspace learning is a classical but powerful method in visual tracking [27]. In this paradigm, the targets on successive frames are considered to reside in a low-dimensional subspace. The principal component analysis (PCA) approach is used to learn the basis vectors, and the target is located in terms of representation accuracy (reconstruction errors over the learned basis vectors). The underlying assumption of this method is that the target can be well represented (reconstructed) by the learned subspace, leading to small and dense reconstruction errors. From stochastic point of view, the reconstruction errors are assumed to obey a Gaussian distribution with zero mean and small variances. Subspace learning has been demonstrated to be effective in some challenging situations, *e.g.*, illumination variations and pose changes [12, 6]. However, it is unstable in the case of occlusions, because the reconstruction errors that may be large and sparse cannot be explained by Gaussian distribution. Motivated by the latest study, robust PCA [3], we decompose the reconstruction error as two additive errors: one that is small and dense, and the other that is large and sparse. The former is used to maintain the assumption of low-dimensional subspace, and the latter is used to compensate the outliers, *e.g.*, occlusions. Similar solutions are also used in the recent tracking studies [33, 37, 34], achieving impressive tracking results.

Previous work [27, 13, 33, 34, 37] constructs a subspace where only the previously obtained targets (*i.e.*, the targets located on historical frames) reside. In our tracking model, because of the discriminative augmentation, we need to consider both the target and background. For this reason, we construct a subspace where the previously obtained targets and the background patches reside. This is also one of the major differences between our work and previous subspace-based methods. We densely sample the image patches around the latest obtained target and use them as background samples. It indicates that the background patches are still very similar to the recently obtained targets. Thus, we assume that the *recently* obtained targets and the *neighboring* background patches reside in the same low-dimensional subspace. Note that because the background patches are involved, to maintain the assumption of low-dimensional subspace, we only represent the recently obtained targets. Such *temporal locality* is also a difference from previous subspace-based methods that represent all the obtained targets. Fig. 2 illustrates the relationship between classical and our subspace models. We also note that the dimension of our subspace is higher than that of the classical subspace, but still much lower than that of the entire observation space.

In addition, it is very critical to determine an appropriate dimension for the learned subspace. Too low dimension may lead to weak representation capability, whereas too high dimension may result in weak generalization capability. Most of previous work uses an empirically fixed dimension, or imposes a hard threshold on the principal components of training samples. In this work, under the assumption of low-dimensional subspace, the sample matrix, of which each column denotes a sample used to learn the subspace, is considered to be low-rank. Thus, we refer to $rank$-minimization to construct our subspace with an adaptive dimension. Instead of directly evaluating the basis vectors, we minimize the rank of the sample matrix. The rank value is used as the dimension of our subspace, leading to a trade-off between representation and generalization.

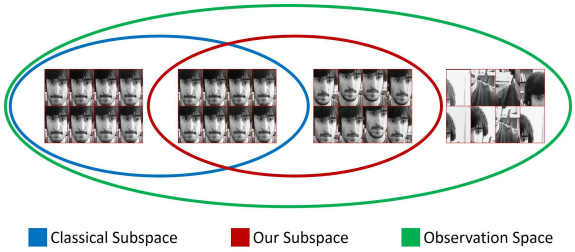


Figure 2. Illustration of the relationship between classical and our subspace models.

1.2. When Representation Meets Discrimination

For either representation or discrimination, the final goal is to locate the target accurately and robustly on each frame, according to representation accuracy or discrimination reliability. In this work, we propose a novel criterion of target location, which takes into account both the representation and discrimination information. It is grounded on the fact that our training samples consist of the recently obtained targets and the neighboring background patches. On one hand, our classifier can effectively distinguish the target from the neighboring background patches. However, it is unreliable to the patches far away from the target due to the lack of training over those samples, *i.e.*, it may recognize those distant background patches as the target. On the other hand, our subspace can successfully exclude the background patches far away from the target according to their large and dense reconstruction errors. However, it cannot separate the target from the neighboring background patches because all of those patches are well-represented by our subspace (have small and sparse reconstruction errors). In brief, the reconstruction errors can successfully recognize whether a testing patch belongs to our subspace or not, and the linear classifier can effectively recognize a testing patch in our subspace as either the target or the background. As a result, we propose the novel criterion of target location, which considers that the patches with both higher discrimination reliability and higher representation accuracy are more likely to be the target.

More importantly, we should note that our strategy of target location effectively avoids the dilemma of sample label ambiguity in discriminative model-based tracking methods, which is addressed in [2]. A robust discriminative tracking model needs to be trained over the samples from entire observation space. However, there are ambiguities in the labels of the background patches close to the target because those patches are very similar to the target. In [2], Babenko *et al*. use the multiple instances learning method to alleviate the ambiguity. In this work, with our subspace and target location strategy, such a dilemma is not involved. Our classifier is trained over only a small number of training samples, which consist of the recently obtained targets and the neighboring background patches, leading to good discriminative performance over the neighboring region, while the background patches far away from the target are dealt with in terms of the generative information (*i.e*., the reconstruction errors).

1.3. Our Contributions

We propose a novel tracking algorithm via a discriminative low-rank learning method, which utilizes two very simple approaches but achieves state-of-the-art performance.

- We propose a discriminative subspace to represent both the recently obtained targets and the neighboring background patches, and simultaneously augment its discriminative performance by training a linear classifier within the joint learning approach. The outliers (*e.g*., occlusions in tracking) are dealt with by using sparse learning during the subspace construction. The dimension of the subspace is adaptively determined via rank-minimization.
- We propose a novel criterion of target location, which takes into account both the discrimination reliability and representation accuracy. Further, with our subspace and target location strategy, the dilemma of sample label ambiguity in discriminative tracking model is effectively alleviated.

2. Related Work

Subspace-based tracking methods have been extensively studied in recent years. Ross *et al*. [27] introduce incremental subspace learning to visual tracking, where the temporally obtained targets are assumed to reside in a low-dimensional subspace. Kwon and Lee [13] apply sparse PCA to formulate tracking. Wang and Lu [32] formulate visual tracking as a subspace learning problem with a possibility continuous outlier model and solve it by using max-flow/min-cut method. Under the subspace assumption, Sui and Zhang [29] propose a locally structured Gaussian Process and cast tracking as a regression problem. As we addressed above, subspace learning is essentially sensitive to occlusions. For this reason, some ad hoc strategies for occlusion handling need to be used with subspace learning together. One popular approach is sparse representation [35].

Mei and Ling [21] introduce sparse representation to visual tracking. In their method, the current target is represented as a linear combination of a few previously obtained targets, and the occlusions are absorbed by ad hoc designed trivial templates. This paradigm has shown its strength to deal with partial occlusions. However, its major problem is the expensive computational cost. Inspired by [21], extensive studies exploiting the subspace model by using a sparse additive error are proposed, in order to improve the robustness. Wang *et al*. [34] use the subspace model to represent the target and impose sparsity on the residual errors to deal with occlusions. Zhang *et al*. [38] exploit both the subspace and sparse structures by a low-rank and sparse representation, and model occlusions by a sparse additive error term.

Recently, the trackers based on discriminative model achieve many impressive tracking results. Kalal *et al*. [11] propose a detection-based paradigm for visual tracking, which trains a binary classifier from labeled and unlabeled examples. Babenko *et al*. [2] apply multiple instances learning method to alleviate the ambiguity among the sample labels. Hare *et al*. [7] propose a structured output SVM method for visual tracking. Also, Oron *et al*. [23] leverage discriminative generative framework [16, 20, 5] to solve tracking task, achieving impressive tracking performance.

3. Our Approach

Our tracking algorithm is conducted within particle filtering framework [9, 1]. Each particle corresponds to a ROI in a frame image. The ROI is defined by a motion state variable

$$\mathbf{s} = \{x, y, \sigma\} \tag{1}$$

where x and y denote its 2D position in the frame image, and σ denotes its scaling coefficient. During tracking, numerous motion state variables are predicted on each frame according to the previously obtained targets. The corresponding ROIs are cropped out from the frame image in terms of their motion state variables, and used as the target candidates on this frame. We normalize these ROIs to the same size and stack them into column vectors, respectively. We call such a column vector the *candidate*. The candidate evaluated to be the best according to our criterion of target location is determined as the target, and marked by a bounding box in the frame image according to its motion state variable.

3.1. Discriminative Low-Rank Learning

On the t-th frame, let the matrix $\mathbf{Y} = [\mathbf{y}_{t-n}, \ldots, \mathbf{y}_{t-1}]$ denote the n recently obtained targets, of which each column $\mathbf{y}_i$ corresponds to the target located on the i-the frame.

We densely sample a number of background patches around the latest target $\mathbf{y}_{t-1}$ by shifting the 2D position of $\mathbf{y}_{t-1}$ with small distances along different directions. Then, we normalize these background patches to the same size as the target and stack them into column vectors, respectively. We call these column vectors the background samples, and denote them by the matrix $\mathbf{B}$, of which each column denotes a background sample.

Due to the dense sampling, the background samples are very similar to the recently obtained targets. Thus, we assume that the recently obtained targets $\mathbf{Y}$ and the background samples $\mathbf{B}$ reside in a low-dimensional subspace. It indicates that the sample matrix $\mathbf{X} = [\mathbf{Y}, \mathbf{B}]$ has small rank value (*i.e.*, low-rank). To ensure the robustness of this subspace, we further assume that there exist some outliers in the samples $\mathbf{X}$. Thus, we compensate these outliers by a sparse additive residual error term. As a result, the subspace can accurately represent both the target and background.

Meanwhile, the subspace is also expected to discriminatively represent the target and the background. Thus, we simultaneously train a linear classifier to separate them during the subspace construction. To this end, the subspace can represent the target and background accurately and distinguish them reliably. We cast our goals discussed above into the following joint learning problem:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{E},\mathbf{w},b} rank\left(\mathbf{A}\right) + \lambda \left\|\mathbf{E}\right\|_0 + \mu \left\|\mathbf{w}\right\|_2^2 \\ &s.t. \begin{cases} \mathbf{X} = \mathbf{A} + \mathbf{E} \\ \mathbf{z} = \mathbf{w}^T\mathbf{A} + b\mathbf{1} \end{cases} \end{aligned} \tag{2}$$

where the low-rank matrix $\mathbf{A}$ denotes the reconstructions of the samples $\mathbf{X}$ over the learned subspace, the sparse matrix $\mathbf{E}$ denotes the reconstruction errors, $\{\mathbf{w}, b\}$ denotes the linear classifier, the vector $\mathbf{z}$ denotes the sample labels for $z_i \in \{+1, -1\}$, $\mathbf{1}$ denotes the column vector of which each entry is 1, $rank\left(\mathbf{A}\right)$ calculates the rank value of $\mathbf{A}$, $\left\|\mathbf{E}\right\|_0$ counts the non-zeros of $\mathbf{E}$, and $\lambda > 0$, $\mu > 0$ are the weight parameters. We set λ to the value recommended by [17] and $\mu = 1$ by referring to [21]. Note that Eq. (2) is a NP-hard problem because it simultaneously involves $rank$- and ℓ_0-minimizations. We relax them to their convex conjugates, $trace$- and ℓ_1-minimizations. It also should be note that although the convex relaxations are leveraged, the obtained problem is still non-convex due to the relationship between $\mathbf{A}$ and $\mathbf{w}$. Fortunately, this problem is convex if we fix either of the two variables. Thus, we can develop an iterative algorithm to solve the problem. To this end, we refer to inexact augmented Lagrange multiplier (IALM) method [17] to develop the iterative algorithm. Due to text length limit, we address this iterative algorithm in the supplementary document along with this submission.

Note that the subspace can be seen as a latent variable in Eq. (2), which is indirectly characterized by the reconstructed samples $\mathbf{A}$. To explicitly obtain the learned subspace, we need to simply apply singular value decomposition (SVD) to $\mathbf{A}$:

$$\left[\mathbf{U}, \mathbf{S}, \mathbf{V}^T\right] = svd\left(\mathbf{A}\right) \tag{3}$$

where the orthogonal matrix $\mathbf{U}$ is used as the basis matrix of the learned subspace, and $\mathbf{S}\mathbf{V}^T$ is used as the corresponding subspace representation. Further, to reduce the learned subspace to an appropriate dimension, we use the first r columns of $\mathbf{U}$ as the basis vectors of our subspace, denoted by the matrix $\mathbf{P}$, where $r = rank\left(\mathbf{A}\right)$:

$$\mathbf{P} = \mathbf{U}_{1:rank(\mathbf{A})} \tag{4}$$

Given a candidate, we reconstruct it over the learned subspace $\mathbf{P}$, and its reconstruction (*i.e.*, the noise-free counterpart) can be successfully classified as either the target or the background by the linear classifier $\{\mathbf{w}, b\}$. In this work, our linear classifier outputs the *classification reliability*, instead of classification labels, as the metric for the likelihood of a candidate to be the target or the background.

For a candidate $\mathbf{c}$, the classification reliability is defined as

$$g\left(\mathbf{c}; \mathbf{P}, \mathbf{w}, b\right) = \mathbf{w}^T\mathbf{P}\mathbf{q} + b \tag{5}$$

where $\mathbf{q}$ denotes the representation of $\mathbf{c}$ over the subspace $\mathbf{P}$, which is found by

$$\begin{aligned} &\min_{\mathbf{q},\mathbf{e}} \left\|\mathbf{e}\right\|_0 \\ &s.t.\ \mathbf{c} = \mathbf{P}\mathbf{q} + \mathbf{e} \end{aligned} \tag{6}$$

where the sparse vector $\mathbf{e}$ denotes the reconstruction errors. Note that Eq. (6) presents an ℓ_0-minimization problem and many algorithms can be used to solve it, *e.g.*, IALM [17] (our choice), OMP [24] and LASSO [30].

3.2. Target Location

On the t-the frame, given all the previously obtained targets from the first to $(t-1)$-th frame, denoted by $\mathbf{y}_{1:t-1}$, the motion state of a candidate on the t-th frame, denoted by $\mathbf{s}_t$, is predicted by maximizing the posterior density

$$p\left(\mathbf{s}_t|\mathbf{y}_{1:t-1}\right) = \int p\left(\mathbf{s}_t|\mathbf{s}_{t-1}\right) p\left(\mathbf{s}_{t-1}|\mathbf{y}_{1:t-1}\right) d\mathbf{s}_{t-1} \tag{7}$$

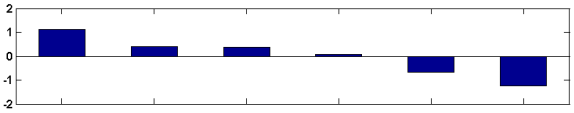
where $p\left(\mathbf{s}_t|\mathbf{s}_{t-1}\right)$ denotes the motion model. Then, the candidate $\mathbf{c}$ is obtained according to the predicted motion state $\mathbf{s}_t$ and the posterior density is updated by

$$p\left(\mathbf{s}_t|\mathbf{c}, \mathbf{y}_{1:t-1}\right) = \frac{p\left(\mathbf{c}|\mathbf{s}_t\right) p\left(\mathbf{s}_t|\mathbf{y}_{1:t-1}\right)}{p\left(\mathbf{c}|\mathbf{y}_{1:t-1}\right)} \tag{8}$$

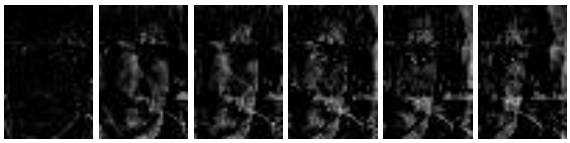
where $p\left(\mathbf{c}|\mathbf{s}_t\right)$ denotes the observation model. Thus, given the set of the candidates $\mathcal{C}$, the target on the t-th frame, denoted by $\mathbf{y}_t$, is found by

$$\mathbf{y}_t = \arg\max_{\mathbf{c}\in\mathcal{C}} p\left(\mathbf{s}_t|\mathbf{c}, \mathbf{y}_{1:t-1}\right) \tag{9}$$

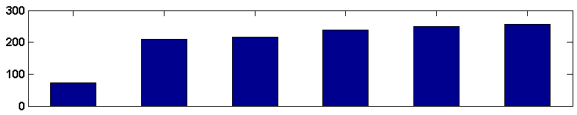
(a) six target candidates



(b) classification reliability $g(\mathbf{c};\mathbf{P},\mathbf{w},b)$



(c) reconstruction errors



(d) magnitude of reconstruction errors $\|\mathbf{e}\|_1$

Figure 3. Six target candidates are shown in (a) and their classification reliability values are shown in (b). The reconstruction errors of the six target candidates are shown in (c), where the darker pixel indicates the smaller value. The corresponding magnitudes of the reconstruction errors are shown in (d).

In this work, the motion model is defined as a Gaussian distribution,

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}) \sim \mathcal{N}(\mathbf{s}_t|\mathbf{s}_{t-1},\mathbf{\Sigma}) \tag{10}$$

where the covariance $\mathbf{\Sigma}$ is a diagonal matrix, of which the diagonal entries denote the variances of x, y and σ in Eq. (1), respectively. The observation model $p(\mathbf{c}|\mathbf{s}_t)$ measures the likelihood to be the target of the candidate $\mathbf{c}$ with the motion state $\mathbf{s}_t$.

On one hand, a good candidate is expected to have the classification reliability as close as possible to $+1$ (label of the target) from the discriminative perspective. It indicates that this candidate is more likely to be a target. Fig. 3(b) shows an illustration of the classification reliability values of the six candidates shown in Fig. 3(a).

On the other hand, a good candidate is also expected to be with the reconstruction error as small as possible from the generative viewpoint. It indicates that the candidate is represented more accurately by the learned subspace. Fig. 3(c) and 3(d) show the reconstruction errors of the six candidates qualitatively and quantitatively, respectively.

To this end, given the learned subspace $\mathbf{P}$ and the linear classifier $\{\mathbf{w}, b\}$, we define the observation model of a candidate $\mathbf{c}$ with the motion state $\mathbf{s}_t$ as

$$p(\mathbf{c}|\mathbf{s}_t) \propto \exp\left\{-\frac{1}{l}\left(|1-g(\mathbf{c};\mathbf{P},\mathbf{w},b)| + \rho\delta(\mathbf{c};\mathbf{P})\right)\right\} \tag{11}$$

where $\delta(\mathbf{c};\mathbf{P}) = \|\mathbf{e}\|_1$ and the reconstruction error $\mathbf{e}$ is obtained from Eq. (6), $\rho > 0$ balances the importance between classification reliability and representation accuracy, and $l > 0$ is the scale parameter. Note that ρ is highly dependent on $\mathbf{c}$ and required to be tuned carefully. To avoid to tune it, we reformulate the observation model as

$$p(\mathbf{c}|\mathbf{s}_t) \propto \exp\left\{-\frac{1}{l}\left(g_{\mathbf{c}} + \delta_{\mathbf{c}}\right)\right\} \tag{12}$$

where $g_{\mathbf{c}}$ and $\delta_{\mathbf{c}}$ denote the normalized classification reliability and reconstruction error of the candidate $\mathbf{c} \in \mathcal{C}$, respectively, and are found by

$$\begin{aligned} g_{\mathbf{c}} &= \frac{|1-g(\mathbf{c};\mathbf{P},\mathbf{w},b)|}{\left\| \left[|1-g(\mathbf{c}_i;\mathbf{P},\mathbf{w},b)|\right]_{\mathbf{c}_i\in\mathcal{C}} \right\|_2} \\ \delta_{\mathbf{c}} &= \frac{\delta(\mathbf{c};\mathbf{P})}{\left\| \left[\delta(\mathbf{c}_i;\mathbf{P})\right]_{\mathbf{c}_i\in\mathcal{C}} \right\|_2} \end{aligned} \tag{13}$$

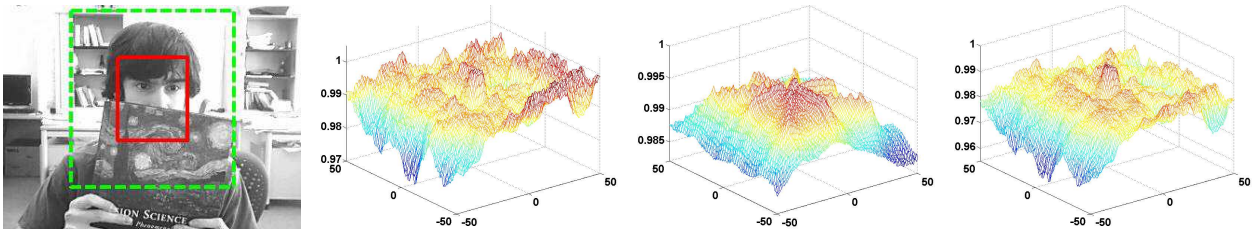
3.3. Update Scheme

During tracking, as the appearance of the target varies, we need to dynamically update the learned subspace $\mathbf{P}$ and the linear classifier $\{\mathbf{w}, b\}$ to capture the latest appearance changes. To maintain the requirement of low dimension, the subspace is required to be learned over the recently obtained targets $\mathbf{Y}$ and the neighboring background samples $\mathbf{B}$. For this reason, we maintain the fifty most recently obtained targets in $\mathbf{Y}$ and update them every frame by replacing the oldest obtained target with the latest obtained target. Thus, $\mathbf{Y}$ works like a first-in-first-out (FIFO) buffer. Meanwhile, because the initially obtained target is the most informative sample, we always keep it in $\mathbf{Y}$. The background samples $\mathbf{B}$ are replaced completely every frame by the samples densely sampled around the latest obtained target. Moreover, as a trade-off between accuracy and efficiency, we re-train the subspace and the linear classifier every ten frames over the training samples $\mathbf{X} = [\mathbf{Y}, \mathbf{B}]$.

3.4. Discussion

Note that the observation model in Eq. (12) defines a criterion of target location by integrating both generative and discriminative information. To make this point more clear, we take an example to explain our target location method. As shown in Fig. 4, we analyze the likelihood to be the target of the ROIs centered at every pixel within the search region in a frame image. The maximum likelihood is expected to appear at the search region center.

Because the subspace and the linear classifier are trained over the recently obtained targets and the neighboring background samples, the linear classifier can output accurate



(a) frame image (b) classification reliability (c) reconstruction error (d) our criterion

Figure 4. Illustration of our criterion of target location. A frame image is shown in (a), where the target is marked by the red (solid) box, and the search region is marked by the green (dashed) box. The likelihood values to be the target of the ROIs centered at every pixel within the search region in terms of classification reliability, reconstruction error and our criterion are plotted in (b), (c) and (d), respectively. The maximum likelihood is expected to appear at the center of each plot. The colder color indicates the smaller value in (b), (c) and (d).

classification reliability for the candidates that can be well represented by the learned subspace. However, it is unstable to the candidates that are poorly represented by the learned subspace. It can be clearly seen from Fig. 4(b) that the ROIs close to the search region center have accurate classification reliability, and that in contrast the ROIs far away from the search region center obtain unstable classification reliability.

Also, the well-represented candidates have smaller and sparser reconstruction errors than the poorly-represented candidates. Thus, the reconstruction errors can be used to exclude the poorly-represented candidates. However, it cannot reliably distinguish the target from the neighboring background among the well-represented candidates. From Fig. 4(c), it can be seen that the ROIs close to the search region center have large likelihood to be the target (small and sparse reconstruction errors) and the ROIs far away from the search region center have small likelihood (large and dense reconstruction errors). Note that the likelihood exhibits some directionality in Fig. 4(c). This is because we sample the background patches along several specific directions.

As a result, we propose the criterion of target location according to the above analysis. The reconstruction errors can be used to choose the well-represented candidates in terms of the magnitude and sparsity. Meanwhile, the classifier can reliably distinguish the target from the background among these well-represented candidates in terms of the classification reliability. Clearly, it can be seen from Fig. 4(d) that: 1) although some poorly-represented candidates have large classification reliability, they are punished by the reconstruction errors, leading to small likelihood to be the target; and 2) the likelihood of the well-represented candidates is dominated by the classification reliability, leading to an reliable discrimination between the target and the background.

More importantly, to ensure the low-dimensional subspace assumption, we note that two rules for samples generation need to be maintained: 1) temporal locality of positive samples, *i.e.*, only select the recently obtained targets, and 2) spatial locality of negative samples, *i.e.*, only select the neighboring background patches.

4. Experiments

Our tracker is implemented in MATLAB on a PC with an Intel Core 2.8GHz processor. The average running speed is 3 frames per second. Solving Eq. (2) needs 0.5 second. On each frame, it costs 0.2 second to solve Eq. (6) for all the candidates. The colorful pixels on each frame are converted to gray scale values and normalized to $[0, 1]$. 400 candidates are generated on each frame and their corresponding ROIs are normalized to 20×20 pixels. The covariance parameter of the motion model in Eq. (10) is set to $\mathbf{\Sigma} = diag\{3, 3, 0.01\}$. The background sample matrix $\mathbf{B}$ consists of 48 columns that respectively correspond to the 48 background patches with translations of $\{7, 9, 11, 13, 15, 17\}$ pixels from the latest target along eight directions that uniformly distributed within $[0°, 360°)$. In Eq. (12), l is set to 10^{-4}.

4.1. Competing Trackers

Our tracker is compared to the other sixteen state-of-the-art methods, involving subspace learning, sparse representation and/or discriminative learning. Because there are no available source codes of LRST, we implement it by ourselves in terms of the corresponding paper [38]. The other fifteen competing trackers are publicly provided by the authors. The parameters of the competing trackers are tuned carefully to obtain their best performance.

4.2. Data Description and Evaluation Criteria

Our tracker is evaluated on a popular benchmark database, Wu *et al*. [36]'s benchmark, which includes fifty challenging video sequences. These video sequences include various complicated factors, *e.g.*, illumination variation, occlusion, non-rigid deformation, cluttered background, and in-plane/out-of-plane rotation. On each frame

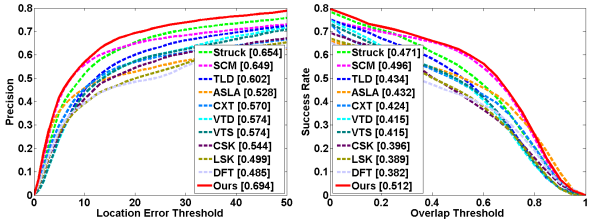


Figure 5. Performance of our tracker and the top ten trackers in [36] on all the 50 video sequences of Wu *et al*.'s benchmark.

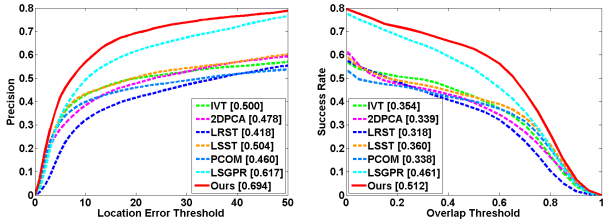


Figure 6. Performance of our and other six subspace-based trackers on all the 50 video sequences of Wu *et al*.'s benchmark.

of the fifty video sequences, the target is labelled manually and used as ground truth for quantitative evaluations.

Two criteria are used in this work to quantitatively evaluate the performance of our tracker:

- *Precision*. The percentage of frames where location errors are less than a threshold. The location error is defined as the distance (in pixel) between the centers of tracking and ground truth bounding boxes.

- *Success Rate*. The percentage of frames where overlap rates are greater than a threshold. The overlap rate on a frame is defined as $\frac{A_T \cap A_G}{A_T \cup A_G}$, where A_T and A_G denote the areas of tracking and ground truth bounding boxes, respectively.

4.3. Comparison against State-of-the-Art Trackers

We report the precisions and success rates of our tracker and the top ten trackers in [36] on the fifty video sequences of Wu *et al*.'s benchmark, including Struck [7], SCM [39], TLD [11], ASLA [10], CXT [4], VTD [13], VTS [14], CSK [8], LSK [18], DFT [28], as shown in Fig. 5. The quantitative results are also shown in the legend of each sub-figure. It can be seen that our tracker performs favourably against the ten state-of-the-art trackers on Wu *et al*.'s benchmark.

Also, we compare our tracker against other six subspace-based trackers on the fifty video sequences of Wu *et al*.'s benchmark, including IVT [27], 2DPCA [31], LRST [38], LSST [33], PCOM [32], LSGPR [29]. The results are reported in Fig. 6. It can be seen that our tracker outperforms the six subspace-based trackers on Wu *et al*.'s benchmark.

For more thorough evaluation of our tracker, we also analyze the performance of our tracker in different challenging situations, *e.g*., illumination variation and occlusion, and the results are shown in Fig. 7.

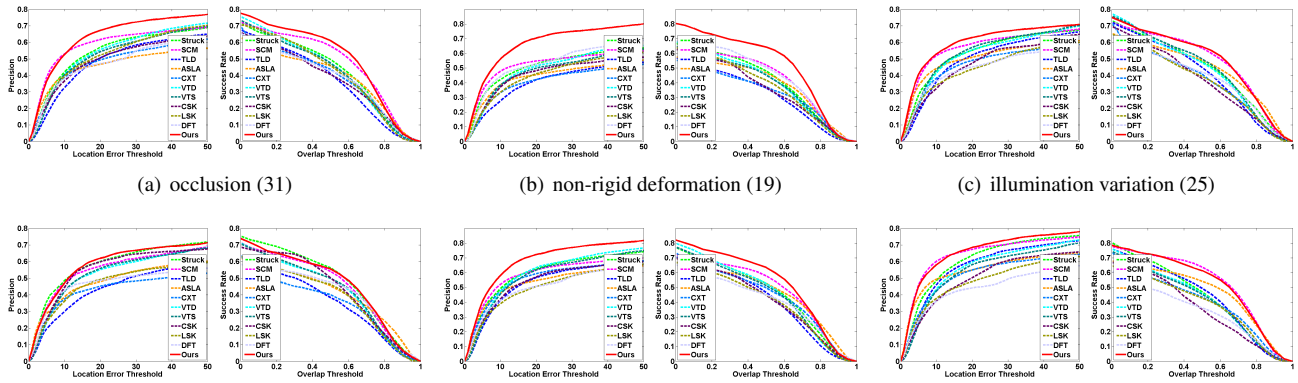
Occlusion. In the case of occlusion, the target is occluded by some similar and/or dissimilar objects. Occlusion may easily lead to tracking failure because the target disappears partially or entirely for a period. From the results shown in Fig. 7(a), it can be seen that our tracker is robust against to occlusion and obtains good tracking results. It benefits from the facts that 1) the sparse reconstruction errors can absorb the occlusion during our subspace learning, such that the learned subspace only acquires the non-occluded information of the target; and 2) the good discriminative capability of the learned subspace can accurately separate the target from the background. The competing trackers using sparse reconstruction errors for occlusion handling, *e.g*., SCM and LSK, and the competing trackers using discriminative tracking model, *e.g*., Struck, also achieve good tracking results on some video sequences in this case.

Non-Rigid Deformation. The motion of the target may cause non-rigid deformations in the appearance. From the results shown in Fig. 7(b), it can be seen that our tracker obtains good performance in this case. This is attributed to the facts that 1) the small deformation, which causes small reconstruction errors, is effectively dealt with by the subspace learning; and 2) the large deformation, which causes large reconstruction errors, is compensated by using the sparsity constraint on the reconstruction errors.

Illumination Variation. In this case, the illumination of the scene changes drastically, leading to significant changes in the appearance of the target. From the results shown in Fig. 7(c), it can be seen that our tracker obtains good results in this case. This is attributed to that the subspace learning is effective to deal with illumination change. Note that the adaptive dimension reduction of our subspace learning also makes our tracker more stable in this case. It can also be seen that some subspace learning based trackers, *e.g*., VTD and VTS, also obtain good tracking performance.

Background Clutter. In this situation, the tracker is distracted by the cluttered background. Thus, the tracker that considers the difference between the target and the background information may be more effective in this case. From the results shown in Fig. 7(d), it can be seen that our tracker performs well in this case. This is attributed to that our tracker has good discriminative capability, which can reliably distinguish the target from the background. As we analyzed above, the competing trackers that considers the background, *e.g*., Struck, CSK and SCM, also obtain good tracking results in this case.

Out-of-Plane Rotation. The motion of either the target or the camera may cause out-of-plane rotations in the appearance of the target. From the results shown in Fig. 7(e), it can be seen that our tracker performs well in this case. On one hand, the temporal locality of our subspace (only using the recently obtained targets) is effective to describe the ap-



(a) occlusion (31) (b) non-rigid deformation (19) (c) illumination variation (25)

(d) background clutter (22) (e) out-of-plane rotation (39) (f) scale variation (28)

Figure 7. Performance of our tracker and the top ten ranking trackers in [36] in different challenging cases. In the caption of each sub-figure, the number in parentheses denotes the number of the video sequences in the corresponding case.

pearance changes caused by out-of-plane rotations. On the other hand, the linear classifier can successfully separate the target with out-of-plane rotations from the background.

Scale Variation. In this case, the scale of the appearance of the target on successive frames varies over time, such that the tracker may result in inaccurate tracking results. Because we take into account the scale change of the target in the motion state, as shown in Eq. (1), it can be seen from the results shown in Fig. 7(f) that our tracker is insensitive to scale change and obtains good performance.

Overall, from the above results, it can be seen that our tracker performs favourably against the sixteen competing trackers on Wu *et al.*'s benchmark.

4.4. Effectiveness of Discrimination

One contribution of our work is to include neighboring background patches to augment the discriminative capability of the learned subspace. Thus, we demonstrate the effectiveness of the discrimination. As shown in Fig. 8, although the dimension of the learned subspace is reduced to be very low (only three-dimensional), the samples are almost linearly separatable. It indicates that the learned subspace has good discriminative capability for visual tracking.

5. Limitations

Within particle filtering framework, the variances of translations are responsible to the searching range for candidates. In this work, the variances of translations in both x and y directions are set to 3 pixels. It indicates that if the target moves very fast between two consecutive frames, our tracker may lose the target with a high possibility. In fact, we are facing the dilemma of the trade-off between accuracy and efficiency. On one hand, a large number of candidates are expected to densely generate around possible target locations in a wide range regions, in order to locate the target as accurately as possible. On the other hand, the number

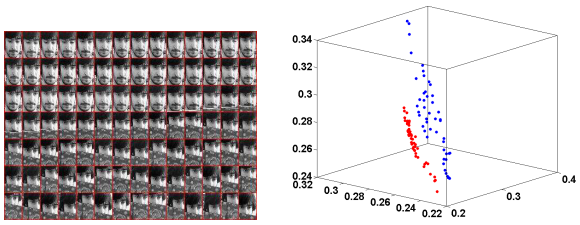


Figure 8. Effectiveness of discriminative low-rank learning. The positive (first fifty) and negative (the rest) samples are shown in the left. The reconstructed positive (red) and negative (blue) samples over the first three basis vectors are shown in the right.

of candidates is required as small as possible, because the computational cost is linearly proportional to the number of candidates. Thus, considering the balance between accuracy and efficiency, we have to shrink the searching range (*i.e.*, use small variances of translations, *e.g.*, 3 pixels) to ensure that adequate candidates are generated in the regions where the next target will appear with higher possibilities.

6. Conclusion

We have proposed a novel tracking algorithm that simultaneously exploits the advantages of both subspace learning and discriminative learning, aiming to alleviate the tracking drift problem in various challenging situations. A large number of experiments have been conducted and the results have shown that 1) the proposed discriminative low-rank learning leads to an effective and robust tracker; and 2) the criterion of target location facilitates to alleviate the tracking drift problem. Both the qualitative and quantitative evaluations on numerous challenging video sequences have demonstrated that our tracker performs favourably against many other state-of-the-art trackers.

Acknowledgement: This work was supported by the National Natural Science Foundation of China under Grant 61172125 and Grant 61132007.

References

[1] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing (TSP)*, 50(2):174–188, 2002. 3

[2] B. Babenko, S. Member, M.-H. Yang, and S. Member. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(8):1619–1632, 2011. 3

[3] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, May 2011. 2

[4] T. B. Dinh, N. Vo, and G. Medioni. Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments. In *CVPR*, 2011. 7

[5] B. Ghanem and N. Ahuja. A Probabilistic Framework for Discriminative Dictionary Learning. *arXiv:1109.2389v1 [cs.CV]*, pages 1–10, 2011. 3

[6] G. D. Hager and P. N. Belhumeur. Real-Time Tracking of Image Regions with Changes in Geometry and Illumination. In *CVPR*, 1996. 2

[7] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011. 3, 7

[8] F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In *ECCV*, 2012. 7

[9] M. Isard. CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28, 1998. 3

[10] X. Jia, H. Lu, and M.-H. Yang. Visual Tracking via Adaptive Structural Local Sparse Appearance Model. In *CVPR*, 2012. 7

[11] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010. 3, 7

[12] D. J. Kriegmant, E. Engineering, and N. Haven. What is the Set of Images of an Object Under All Possible Lighting Conditions? In *CVPR*, 1996. 2

[13] J. Kwon and K. Lee. Visual tracking decomposition. In *CVPR*, 2010. 2, 3, 7

[14] J. Kwon and K. M. Lee. Tracking by Sampling Trackers. In *ICCV*, 2011. 7

[15] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled Hybrids of Generative and Discriminative Models. In *CVPR*, 2006. 1

[16] R.-S. Lin, D. Ross, J. Lim, and M.-H. Yang. Adaptive Discriminative Generative Model and Its-Applications. In *NIPS*, 2004. 3

[17] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report*, pages 1–23, 2010. 4

[18] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and K-selection. In *CVPR*, 2011. 7

[19] J. Mairal, F. Bach, and J. Ponce. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008. 1

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009. 3

[21] X. Mei and H. Ling. Robust visual tracking using L1 minimization. In *ICCV*, 2009. 3, 4

[22] A. Y. Ng and M. I. Jordan. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. In *NIPS*, 2001. 1

[23] S. Oron, A. Bar-hillel, and S. Avidan. Extended Lucas-Kanade Tracking. In *ECCV*, 2014. 3

[24] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, 1993. 4

[25] D.-S. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *CVPR*, 2008. 1

[26] R. Raina and A. Y. Ng. Self-taught Learning : Transfer Learning from Unlabeled Data. In *ICML*, 2007. 1

[27] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2007. 2, 3, 7

[28] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012. 7

[29] Y. Sui and L. Zhang. Visual Tracking via Locally Structured Gaussian Process Regression. *IEEE Signal Processing Letters*, 22(9):1331–1335, 2015. 3, 7

[30] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. 4

[31] D. Wang and H. Lu. Object tracking via 2DPCA and L1-regularization. *IEEE Signal Processing Letters*, 19(11):711–714, 2012. 7

[32] D. Wang and H. Lu. Visual Tracking via Probability Continuous Outlier Model. In *CVPR*, 2014. 3, 7

[33] D. Wang, H. Lu, and M.-H. Yang. Least Soft-thresold Squares Tracking. In *CVPR*, 2013. 2, 7

[34] D. Wang, H. Lu, and M.-H. Yang. Online object tracking with sparse prototypes. *IEEE Transactions on Image Processing (TIP)*, 22(1):314–325, 2013. 2, 3

[35] J. Wright, Y. Ma, J. Mairal, and G. Sapiro. Sparse representation for computer vision and pattern recognition. *Proceedings of The IEEE*, 98(6):1031–1044, 2010. 3

[36] Y. Wu, J. Lim, and M.-H. Yang. Online Object Tracking: A Benchmark. In *CVPR*, 2013. 6, 7, 8

[37] C. Zhang, R. Liu, T. Qiu, and Z. Su. Robust visual tracking via incremental low-rank features learning. *Neurocomputing*, 131:237–247, May 2014. 2

[38] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In *ECCV*, 2012. 3, 6, 7

[39] W. Zhong, H. Lu, and M.-H. Yang. Robust Object Tracking via Sparsity-based Collaborative Model. In *CVPR*, 2012. 7