Discriminant Tracking Using Tensor Representation with Semi-supervised Improvement

Jin Gao[1], Junliang Xing[1], Weiming Hu[1], and Steve Maybank[2]

[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing
[2]Department of Computer Science, Birkbeck College, London
{jgao10, jlxing, wmhu}@nlpr.ia.ac.cn, sjmaybank@dcs.bbk.ac.uk

Abstract

Visual tracking has witnessed growing methods in object representation, which is crucial to robust tracking. The dominant mechanism in object representation is using image features encoded in a vector as observations to perform tracking, without considering that an image is intrinsically a matrix, or a 2^{nd}*-order tensor. Thus approaches following this mechanism inevitably lose a lot of useful information, and therefore cannot fully exploit the spatial correlations within the 2D image ensembles. In this paper, we address an image as a* 2^{nd}*-order tensor in its original form, and find a discriminative linear embedding space approximation to the original nonlinear submanifold embedded in the tensor space based on the graph embedding framework. We specially design two graphs for characterizing the intrinsic local geometrical structure of the tensor space, so as to retain more discriminant information when reducing the dimension along certain tensor dimensions. However, spatial correlations within a tensor are not limited to the elements along these dimensions. This means that some part of the discriminant information may not be encoded in the embedding space. We introduce a novel technique called semi-supervised improvement to iteratively adjust the embedding space to compensate for the loss of discriminant information, hence improving the performance of our tracker. Experimental results on challenging videos demonstrate the effectiveness and robustness of the proposed tracker.*

1. Introduction

Robust visual tracking is an essential component of many practical computer vision applications such as surveillance, vehicle navigation, and human computer interface. Despite much effort resulting in many novel trackers, tracking generic objects remains a challenging problem because of the intrinsic (*e.g.* deformations, out-of-plane rotations) and extrinsic (*e.g.* partial occlusions, illumination changes, cluttered and moving backgrounds) appearance variations (see a more detailed discussion in [31]).

Many tracking systems construct an adaptive appearance model based on the collected image patches in previous frames. This model is used to find the most likely image patch in the current frame. Image patch representation is crucial to this process. There are many object representation methods proposed for visual tracking. Some tracking approaches [21, 35] adopt holistic gray-scale image-as-vector representation. These also include ℓ_1 minimization based visual tracking approaches [17, 18, 3, 34, 33], which exploit the sparse representation of the image patch. Kwon *et al.* [8, 9] construct multiple basic appearance models by sparse principal component analysis (SPCA) of a set of feature templates (*e.g.* global image-as-vector descriptors of hue, saturation, intensity, and edge information). These representation methods ignore that an image is intrinsically a matrix, or a 2^{nd}-order tensor. Grabner *et al.* [5, 6] use Haar-like features, histograms of oriented gradients (HOGs), and local binary patterns (LBPs) to obtain weak hypotheses for boosting based tracking. In [2, 10] only Haar-like features used, but great improvements are achieved by novel appearance models. Li *et al.* [12, 15] only use HOGs but apply novel appearance models to achieve good results. Adam *et al.* [1] robustly combine multiple patch votes with each image patch represented by only gray-scale histogram features. These representation methods have their own advantages for their specifically designed appearance models. However, a lot of useful information is missed when extracting features. We claim that tensor representation methods (*e.g.* image-as-matrix representation) can retain much more useful information because the original image data structure is preserved.

There have been many studies (*e.g.* [29, 7, 27]) on tensor-based subspace learning, particularly for face recognition. Also, many previous visual tracking approaches use the tensor concept. Some (*e.g.* [13, 25, 24]) conduct PCA in the mode-k flattened matrix; others (*e.g.* [26, 11]) adopt covariance tracking technique [19] in the mode-k flattened matrix. Although these methods take the correlations along different dimensions of the tensor into account, they share a problem that their generative learning based appearance models ignore the influence of the background and consequently suffer from distractions caused by the background regions with similar appearances to foreground objects. Additionally, the dimension reduction based subspace learning methods used in [13, 25, 24] ignore a very important problem proposed in [27]. The problem is that correlations within a tensor are not limited to the elements along certain tensor dimensions. Some part of the discriminant information may not be encoded in the first few dimensions of the derived subspace. This may lead to subspace learning degradations and result in tracking distractions. A direct comparison of results between the methods of IRTSA [13] and ours is given in Section 3.2. Although Yan *et al.* [27] propose to rearrange elements in the tensor to solve the subspace learning degradation problem, the exhaustive element rearranging makes it unsuitable for real-time tracking.

Inspired by these findings, we propose a new discriminant tracking approach which adopts a 2^{nd}-order tensor (image-as-matrix) representation. Unlike the image-as-vector representation methods which mask the underlying high-order structure by transforming the input data into a vector leading to the loss of discriminant information, we consider an image patch as a 2^{nd}-order tensor, where the relationship between the column vectors and that between the row vectors are characterized individually. Then, we embed the target and background tensor samples into two specially designed graphs, so that the object can be effectively separated from the background in the graph embedding framework for dimension reduction. It is noted that, this approach can be extended by using higher-order tensor representation (*e.g.* 3^{rd}-order tensor with a feature vector for each pixel, see [25, 24, 26, 11] for more details), although we only use gray-scale image-as-matrix representation. Because the correlations within a 2^{nd}-order tensor are not limited to the elements along particular columns and rows, the discriminative embedding space derived from dimension reduction may not encode enough of the discriminant information. We improve the classification accuracy of our tensor representation based tracking approach by using the available unlabeled tensor samples. This is called as semi-supervised improvement. By this improvement, we can adjust the discriminative embedding space so that most of the discriminant information is encoded in it.

The improvement is carried out iteratively. At each iteration, a number of unlabeled tensor samples are selected and used to learn a new discriminative embedding space. The learned embedding spaces from different iterations and the one trained using only the labeled samples are combined linearly to form a final adjusted embedding space which encodes most of the discriminant information. That is to say, we make use of the unlabeled samples in an inductive fashion, which is very different from most semi-supervised tracking approaches (*e.g.* [6, 10, 12]), in which all the unlabeled samples are used for training without selection. The new semi-supervised improvement technique adopts a novel strategy to address the two questions: 1) how to select the unlabeled samples; 2) what class labels should be assigned to the selected unlabeled samples. It is also very different from some margin improving techniques, where the unlabeled samples with the highest classification confidences are selected and the class labels that are predicted by the current classifier are assigned to them, as in Self-training [20], ASSEMBLE [4]. These techniques may increase the classification margin, but they do not provide any novel information to adjust the discriminative embedding space. We plug the margin improving technique ASSEMBLE into our system and make a direct comparison with our method in Section 3.1.

2. The Proposed Discriminant Tracking

Figure 1 is an overview of the proposed approach. We elaborate the important components of the proposed approach in this section, in particular the tensor based linear embedding and derivation of the proposed semi-supervised improvement technique. Before all of these, we first review some terminology for tensor operations.

2.1. Terminology for tensor operations

A tensor is a higher order generalization of a vector (1^{st}-order tensor) and a matrix (2^{nd}-order tensor). A tensor is a multilinear mapping over a set of vector spaces. An n^{th}-order tensor is denoted as $\mathcal{A} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_n}$, and its elements are represented by $a_{i_1,\ldots,i_n}$. The inner product of two n^{th}-order tensors is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1,\ldots,i_n=1}^{i_1=m_1,\ldots,i_n=m_n} a_{i_1,\ldots,i_n} b_{i_1,\ldots,i_n},$$

the norm of $\mathcal{A}$ is $\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$, and the distance between $\mathcal{A}$ and $\mathcal{B}$ is $\|\mathcal{A} - \mathcal{B}\|$. In the 2^{nd}-order tensor case, the norm is called the Frobenius norm and written as $\|\mathcal{A}\|_F$.

The mode-k vectors are the column vectors of matrix $\mathbf{A}_{(k)} \in \mathbb{R}^{m_k \times (m_1 m_2 \ldots m_{k-1} m_{k+1} \ldots m_n)}$ that results from *flattening* the tensor $\mathcal{A}$. The inverse operation of mode-k *flattening* is mode-k *folding*, which restores the original tensor $\mathcal{A}$ from $\mathbf{A}_{(k)}$. The mode-k product of a tensor $\mathcal{A} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_n}$ and a matrix $\mathbf{M} \in$

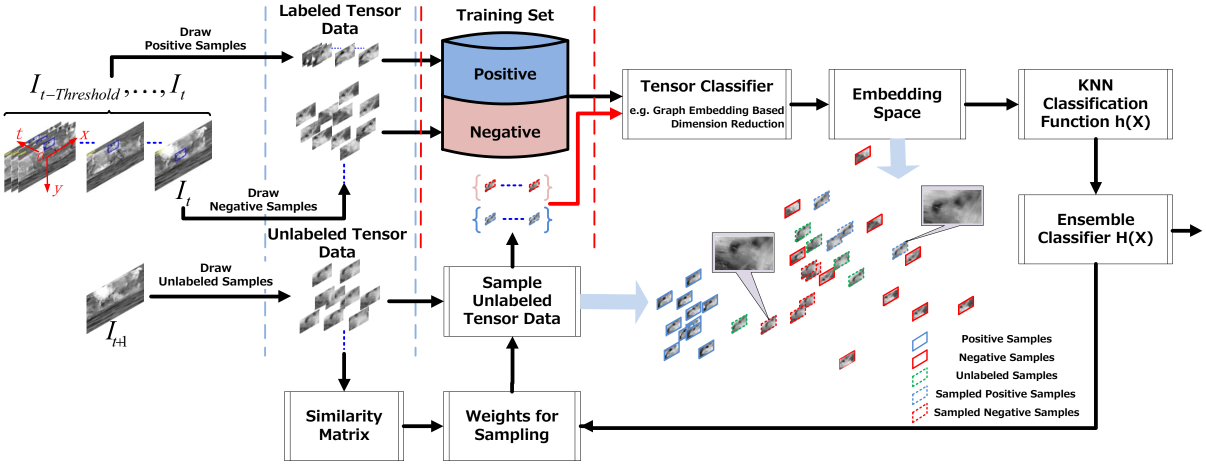


Figure 1: Block diagram of the proposed tracker.

$\mathbb{R}^{l_k\times m_k}$ is denoted by $\mathcal{A}\times_k \mathbf{M}$. Its result is a tensor $\mathcal{C} \in \mathbb{R}^{m_1\times\ldots\times m_{k-1}\times l_k\times m_{k+1}\times\ldots\times m_n}$ whose entries are $c_{i_1,\ldots,i_{k-1},j,i_{k+1},\ldots,i_n} = \sum_{i=1}^{m_k} a_{i_1,\ldots,i_{k-1},i,i_{k+1},\ldots,i_n} \times m_{ji}, j = 1,\ldots,l_k$. The tensor $\mathcal{C}$ can also be computed by matrix multiplication $\mathbf{C}_{(k)} = \mathbf{M}\mathbf{A}_{(k)}$, followed by mode-$k$ *folding*. Note that for tensors and matrices of the appropriate sizes, $\mathcal{A}\times_m \mathbf{U}\times_n \mathbf{V} = \mathcal{A}\times_n \mathbf{V}\times_m \mathbf{U}$ and $(\mathcal{A}\times_n \mathbf{U})\times_n \mathbf{V} = \mathcal{A}\times_n(\mathbf{V}\mathbf{U})$. More details of the tensor algebra are given in [23].

2.2. Tensor based linear embedding

Previous work has demonstrated that the image variations of many objects can be modeled by low dimensional linear spaces. However, the typical algorithms either only consider an image as a high dimensional vector, or can not fully detect the intrinsic local geometrical and discriminative structure of the collected image patches in the tensor form. Then, a particular question arises: how to find an effective linear embedding space approximation to the original nonlinear submanifold embedded in the tensor space. Graph embedding for dimension reduction [28, 22] provides us an innovation to this question in the sense of local isometry.

Generally case: We express the training sample set in the tensor form as $\{\mathcal{X}_i \in \mathbb{R}^{m_1\times\ldots\times m_n}, i = 1,2,\ldots,N\}$. They are sampled from a submanifold $\mathcal{M} \subseteq \mathbb{R}^{m_1\times m_2\times\ldots\times m_n}$. We build two graphs: an intrinsic graph $\mathcal{G}$ and a penalty graph $\mathcal{G}^p$ to model the local geometrical and discriminative structure of $\mathcal{M}$. Let $\mathbf{W}$ and $\mathbf{W}^p$ be the edge weight matrices of $\mathcal{G}$ and $\mathcal{G}^p$. Let $\left\{\mathbf{M}^k \in \mathbb{R}^{l_k\times m_k}, k = 1,2,\ldots,n, l_k < m_k\right\}$ be the transformation matrices that map those N samples to a set of points $\left\{\mathcal{Y}_i \in \mathbb{R}^{l_1\times l_2\times\ldots\times l_n}, i = 1,2,\ldots,N\right\}$ in the low dimensional linear space, where $\mathcal{Y}_i = \mathcal{X}_i\times_1 \mathbf{M}^1\times_2 \mathbf{M}^2\ldots\times_n \mathbf{M}^n$. Then, a reasonable transformation respecting the graph structure can be obtained by solving the following objective function:

$$\left\{\mathbf{M}^{k*}\right\} = \underset{\sum\limits_{i,j}\|\mathcal{Y}_i-\mathcal{Y}_j\|^2 W^p_{ij}=d}{\operatorname{argmin}} \sum_{i,j}\left\|\mathcal{Y}_i-\mathcal{Y}_j\right\|^2 W_{ij} \qquad (1)$$

where $k = 1,2,\ldots,n$ and d is a constant.

2$^{\text{nd}}$-order tensor case: In our proposed tracking approach, we express the training sample set in 2$^{\text{nd}}$-order tensor form as $\{\mathcal{X}_i \in \mathbb{R}^{m_1\times m_2}, i = 1,2,\ldots,N\}$. As shown in Figure 1, we draw positive tensor samples based on the tracking results at previous frames, so the variations of these samples are mainly caused by the object appearance variations over time. We draw negative tensor samples from the surrounding regions of the tracking result in the current frame I_t using a dense sampling method, so the variations of these samples mainly depend on the sampling positions and the background variations. Two-dimensional linear discriminant analysis (2DLDA) [30] has been proposed to detect the discriminative structure of 2$^{\text{nd}}$-order tensor samples, while the intrinsic local geometrical structure of them can not be detected, because 2DLDA does not take the variations of the samples in the same class into account. These variations are often observed in tracking applications. Here we design two graphs to model the local geometrical and discriminative structure of the training tensor samples in our tracking application. Let $y_i \in \{-2,-1,+1,+2\}$ be associated class labels with the training set, where $+2$ indicates the sampled positive samples from unlabeled samples, $+1$ indicates the labeled positive (object) samples, -1 indicates the labeled negative (background) samples, and -2 indicates the sampled negative samples from unlabeled samples, as shown in Figure 1. Let n_c be the number of samples in the class c:$\sum_{c=-2}^{+2} n_c = N$. We also denote $n_- = n_{-2}+n_{-1}, n_+ = n_{+2}+n_{+1}, \hat{n}_- = \sqrt{n_{-1}n_{-2}}$, and $\hat{n}_+ = \sqrt{n_{+1}n_{+2}}$.

Construct the intrinsic graph $\mathcal{G}$: In $\mathcal{G}$, the elements W_{ij}

of $\mathbf{W}$ are defined as follows:

$$W_{ij} = \begin{cases} A_{ij}/n_c, & \text{if } y_i = y_j = c, \\ A_{ij}/\hat{n}_+, & \text{elseif } y_i > 0 \text{ and } y_j > 0, \\ A_{ij}/\hat{n}_-, & \text{elseif } y_i < 0 \text{ and } y_j < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The affinity A_{ij} is defined by the local scaling method in [32]. Without loss of generality, we assume that the data points in $\{\mathcal{X}_i\}_{i=1}^N$ are ordered according to their labels $y_i \in \{-2, -1, +1, +2\}$. When $y_i > 0$ and $y_j > 0$,

$$A_{ij} = \exp\left(-\|\mathcal{X}_i - \mathcal{X}_j\|^2/(\sigma_i\sigma_j)\right), \tag{3}$$

where $\sigma_i = \left\|\mathcal{X}_i - \mathcal{X}_i^{(k)}\right\|$, and $\mathcal{X}_i^{(k)}$ is the kth nearest neighbor of $\mathcal{X}_i$ in $\{\mathcal{X}_j\}_{j=n_-+1}^N$. When $y_i < 0$ and $y_j < 0$,

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|\mathcal{X}_i - \mathcal{X}_j\|^2}{\sigma_i\sigma_j}\right), \text{if } i \in N_k^+(j) \text{ or } j \in N_k^+(i), \\ 0, \text{otherwise,} \end{cases} \tag{4}$$

where $N_k^+(i)$ indicates the index set of the k nearest neighbors of $\mathcal{X}_i$ in $\{\mathcal{X}_j\}_{j=1}^{n-}$, $\sigma_i = \left\|\mathcal{X}_i - \mathcal{X}_i^{(k)}\right\|$, and $\mathcal{X}_i^{(k)}$ is the kth nearest neighbor of $\mathcal{X}_i$ in $\{\mathcal{X}_j\}_{j=1}^{n-}$. The parameter k above is empirically chosen as 7 based on [32].

Construct the penalty graph $\mathcal{G}^p$: In $\mathcal{G}^p$, the elements W_{ij}^p of $\mathbf{W}^p$ are defined as follows:

$$W_{ij}^p = \begin{cases} A_{ij}\left(1/N - 1/n_c\right), \text{if } y_i = y_j = c, \\ A_{ij}\left(1/N - 1/\hat{n}_+\right), \text{elseif } y_i > 0 \text{ and } y_j > 0, \\ A_{ij}\left(1/N - 1/\hat{n}_-\right), \text{elseif } y_i < 0 \text{ and } y_j < 0, \\ 1/N, \text{otherwise,} \end{cases} \tag{5}$$

Note that we add the local isometry into the graph structure. Specifically, we determine the embedding space so that, i) when $y_i \cdot y_j > 0$, the nearby data pairs (large values of A_{ij}) are close and the data pairs far apart (small values of A_{ij}) are not required to be close; ii) when $y_i \cdot y_j < 0$, the data pairs are apart by imposing $1/N$. In addition, when we combine the sampled unlabeled samples with the labeled samples to train a new classification model (see Algorithm 2) for improvement of original classifier, we increase the contribution of the sampled unlabeled samples by two means: i) imposing $1/n_c$, $1/\hat{n}_+$ and $1/\hat{n}_-$ in Eq. (2) and Eq. (5) ($n_{-2} < n_{-1}$, $n_{+2} < n_{+1}$ in general); ii) assigning the labels $+2$ or -2 to the sampled unlabeled samples.

By some tensor operations, solving the objective function Eq. (1) in the 2^{nd}-order tensor case is equivalent to solving the following constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\text{minimize}} \quad & \sum_{i,j} \left\|\mathbf{U}^T\mathcal{X}_i\mathbf{V} - \mathbf{U}^T\mathcal{X}_j\mathbf{V}\right\|_F^2 W_{ij}, \\ \text{subject to} \quad & \sum_{i,j} \left\|\mathbf{U}^T\mathcal{X}_i\mathbf{V} - \mathbf{U}^T\mathcal{X}_j\mathbf{V}\right\|_F^2 W_{ij}^p = d \end{aligned} \tag{6}$$

where $\mathbf{U}^T = \mathbf{M}^1$, $\mathbf{V}^T = \mathbf{M}^2$, and d is a constant. Let $\mathbf{D}$ and $\mathbf{D}^p$ be diagonal matrices, where $D_{ii} = \sum_j W_{ij}$ and $D_{ii}^p = \sum_j W_{ij}^p$. According to [7], the optimization problem Eq. (6) can be further reformulated as either of the following two optimization problems: $\min_{\mathbf{U},\mathbf{V}} \operatorname{tr}\left(\frac{\mathbf{U}^T(\mathbf{D}_V - \mathbf{W}_V)\mathbf{U}}{\mathbf{U}^T\left(\mathbf{D}_V^p - \mathbf{W}_V^p\right)\mathbf{U}}\right)$ or $\min_{\mathbf{U},\mathbf{V}} \operatorname{tr}\left(\frac{\mathbf{V}^T(\mathbf{D}_U - \mathbf{W}_U)\mathbf{V}}{\mathbf{V}^T\left(\mathbf{D}_U^p - \mathbf{W}_U^p\right)\mathbf{V}}\right)$, where
$\mathbf{D}_V = \sum_i D_{ii}\mathcal{X}_i\mathbf{V}\mathbf{V}^T\mathcal{X}_i^T$, $\mathbf{W}_V = \sum_{i,j} W_{ij}\mathcal{X}_i\mathbf{V}\mathbf{V}^T\mathcal{X}_j^T$,
$\mathbf{D}_V^p = \sum_i D_{ii}^p\mathcal{X}_i\mathbf{V}\mathbf{V}^T\mathcal{X}_i^T$, $\mathbf{W}_V^p = \sum_{i,j} W_{ij}^p\mathcal{X}_i\mathbf{V}\mathbf{V}^T\mathcal{X}_j^T$,
$\mathbf{D}_U = \sum_i D_{ii}\mathcal{X}_i^T\mathbf{U}\mathbf{U}^T\mathcal{X}_i$, $\mathbf{W}_U = \sum_{i,j} W_{ij}\mathcal{X}_i^T\mathbf{U}\mathbf{U}^T\mathcal{X}_j$,
$\mathbf{D}_U^p = \sum_i D_{ii}^p\mathcal{X}_i^T\mathbf{U}\mathbf{U}^T\mathcal{X}_i$, $\mathbf{W}_U^p = \sum_{i,j} W_{ij}^p\mathcal{X}_i^T\mathbf{U}\mathbf{U}^T\mathcal{X}_j$.

Algorithm 1 Online Tensor Classifier

```
Input: Training dataset $\{\mathcal{X}_i, i = 1, 2, \ldots, N\}$, and associated
    class labels $y_i \in \{-2, -1, +1, +2\}$.
Output: The linear embedding space projected from original
    nonlinear submanifold with transformation matrices $\mathbf{U}$, $\mathbf{V}$.
1: Initially set $\mathbf{U}$ to the first $l_1$ columns of $\mathbf{I}$ and set iteration $\leftarrow$
   1;
2: Calculate $\mathbf{W}$ and $\mathbf{W}^p$ from Eq. (2) and Eq. (5);
3: for iteration $= 1 \to T$ do
4:     Calculate $\mathbf{D}_U$, $\mathbf{W}_U$, $\mathbf{D}_U^p$, $\mathbf{W}_U^p$;
5:     Compute $\mathbf{V}$ by solving the generalized eigenvector prob-
       lem: $(\mathbf{D}_U^p - \mathbf{W}_U^p)\,\mathbf{v} = \lambda\,(\mathbf{D}_U - \mathbf{W}_U)\,\mathbf{v}$;
6:     Calculate $\mathbf{D}_V$, $\mathbf{W}_V$, $\mathbf{D}_V^p$, $\mathbf{W}_V^p$;
7:     Update $\mathbf{U}$ by solving the generalized eigenvector problem:
       $(\mathbf{D}_V^p - \mathbf{W}_V^p)\,\mathbf{u} = \lambda\,(\mathbf{D}_V - \mathbf{W}_V)\,\mathbf{u}$;
8: end for
9: KNN classifier $h(\mathcal{X})$ is used for classification in the linear
   embedding space for its simplicity.
```

In Algorithm 1, we describe a commonly used computational method to solve the two minimization problems. It has been empirically shown in [29, 7] that the iterative algorithm converges to a satisfactory result within three iterations. To simplify the efficiency analysis of Step 4 and 6 in Algorithm 1, we assume $l_1 = l_2 = l$, and $m_1 = m_2 = m$. The main cost is the calculations of $\mathbf{W}_U$, $\mathbf{W}_V$, $\mathbf{W}_U^p$, $\mathbf{W}_V^p$, each of which has $\mathcal{O}(N^2(m^2 + 2m^2 l))$ floating-point multiplications. However, the sparsity of $\mathbf{W}$ and $\mathbf{W}^p$ make the cost much lower.

2.3. Semi-supervised improvement

Recall that we model the local geometrical and discriminative structure of the nonlinear submanifold $\mathcal{M}$ by building two graphs based on the *manifold assumption*. But the correlations within matrix (in the 2^{nd}-order tensor case) are not limited to the elements along particular columns and rows. The learned discriminative embedding space using only the labeled samples may not encode enough of the discriminant information. So we need to adjust the discriminative embedding space to encode most of the discriminant information. The *cluster assumption* allows us to use the unlabeled data to do this. It states that the data samples with high similarity between them, must share the same label. We define the objective function of the semi-supervised improvement using the *cluster assumption*.

Let $\{\mathcal{X}_i\}_{i=1}^{N_l}$ denote the labeled tensor samples, and $\{\mathcal{X}_i\}_{i=N_l+1}^{N_l+N_u}$ denote the unlabeled ones. Suppose that the labeled ones are labeled by $\mathbf{y}_l = \left(y_1^l, y_2^l, \ldots, y_{N_l}^l\right)$, where each class label y_i^l is either $+1$ or -1. Let $\mathbf{S} = [S_{ij}]_{N_e \times N_e}$ denote the symmetric similarity matrix, where S_{ij} ($S_{ij} \geq 0$) represents the similarity between $\mathcal{X}_i$ and $\mathcal{X}_j$, and $N_e = N_l + N_u$. In this paper, we use the block-division based covariance matrix descriptor to measure the similarity under log-Euclidean Riemannian metric (see [14]). We define the distance between any two samples $\mathcal{X}_i$ and $\mathcal{X}_j$ under this metric $\mathfrak{M}$ as $D_{\mathfrak{M}}(\mathcal{X}_i, \mathcal{X}_j)$. Then the similarity S_{ij} is defined by $S_{ij} = \exp\left(-D_{\mathfrak{M}}^2(\mathcal{X}_i, \mathcal{X}_j)/(\sigma_i\sigma_j)\right)$, where σ_i has the similar definition to that in Eq. (3).

We derive the semi-supervised improvement algorithm using an iterative approach. Let $h^{(0)}(\mathcal{X}) : \mathbb{R}^{m_1 \times m_2} \rightarrow \{-1, +1\}$ denote the tensor classification model that is learned by Algorithm 1 based on only the labeled samples. Let $h^{(t)}(\mathcal{X}) : \mathbb{R}^{m_1 \times m_2} \rightarrow \{-1, +1\}$, which is used to improve the performance of $h^{(0)}(\mathcal{X})$, denote the one that is learned at the t-th iteration by Algorithm 1. Let $H(\mathcal{X}) : \mathbb{R}^{m_1 \times m_2} \rightarrow \mathbb{R}$ denote the linearly combined tensor classification model learned after the first $\hat{T}$ iterations: $H(\mathcal{X}) = \alpha_0 h^{(0)}(\mathcal{X}) + \sum_{t=1}^{\hat{T}} \alpha_t h^{(t)}(\mathcal{X})$, where α_0 and α_t are the combination weights. At the $(\hat{T}+1)$-st iteration, our goal is to find a new tensor classifier $h(\mathcal{X})$ with the combination weight α that can efficiently satisfy the following optimization problem:

$$\begin{aligned}\underset{h(\mathcal{X}),\alpha}{\text{minimize}} \quad & \sum_{i=1}^{N_l}\sum_{j=N_l+1}^{N_e} S_{ij}\exp\left(-2y_i^l(H_j+\alpha h_j)\right) \\ + \quad & C\sum_{i,j=N_l+1}^{N_e} S_{ij}\exp\left(H_i-H_j\right)\exp\left(\alpha(h_i-h_j)\right), \\ \text{subject to} \quad & h\left(\mathcal{X}_i\right) = y_i^l, i = 1, \ldots, N_l\end{aligned} \tag{7}$$

where $H_i \equiv H\left(\mathcal{X}_i\right)$, $h_i \equiv h\left(\mathcal{X}_i\right)$ and C weights the contribution of the inconsistency among the unlabeled data. In fact C is chosen to be $C = N_l/N_u$.

Note that the first term in Eq. (7) measures the inconsistency between labeled and unlabeled samples, and the second one measures the inconsistency among the unlabeled samples. Unlabeled samples which are similar to a labeled sample must share its label, and a set of similar unlabeled samples must share the same label.

By regrouping the terms, we rewrite the optimization problem Eq. (7) as the task of minimizing the function $F = \sum_{i=N_l+1}^{N_e}\left(e^{-2\alpha h_i}p_i + e^{2\alpha h_i}q_i\right)$, where

Algorithm 2 Semi-supervised Improvement

```
Input: Labeled samples $\{\mathcal{X}_i\}_{i=1}^{N_l}$ and associated class labels $\mathbf{y}_l$,
    unlabeled samples $\{\mathcal{X}_i\}_{i=N_l+1}^{N_e}$.
Output: The improved tensor classifier $H(\cdot)$.
Compute the pairwise similarity $S_{ij}$;
Train $h^{(0)}(\mathcal{X})$ by Algorithm 1 based on only $\{\mathcal{X}_i\}_{i=1}^{N_l}$;
Let $H(\cdot) = 0$, and compute $\alpha_0$ using Eqs. (8) (9) (10);
Initialize $H(\mathcal{X}) = \alpha_0 h^{(0)}(\mathcal{X})$;
for $t = 1 \rightarrow \hat{T}$ do
    Compute $p_i$ and $q_i$ for every unlabeled sample by Eqs. (8) (9);
    Sample the unlabeled samples by the weights $|p_i - q_i|$, and
    label them with $z_i = 2 \cdot \text{sign}(p_i - q_i)$;
    Combine the sampled samples with $\{\mathcal{X}_i\}_{i=1}^{N_l}$ to train a new
    tensor classifier $h^{(t)}(\mathcal{X})$ by Algorithm 1;
    Compute $\alpha_t$ using Eq. (10);
    Improve the classification function as $H(\mathcal{X}) \leftarrow H(\mathcal{X}) + \alpha_t h^{(t)}(\mathcal{X})$;
end for
return $H(\mathcal{X}) = \alpha_0 h^{(0)}(\mathcal{X}) + \sum_{t=1}^{\hat{T}} \alpha_t h^{(t)}(\mathcal{X})$.
```

$$p_i = \sum_{j=1}^{N_l} S_{ij}e^{-2H_i}\delta(y_j^l, 1) + \frac{C}{2}\sum_{j=N_l+1}^{N_e} S_{ij}e^{H_j-H_i} \tag{8}$$

$$q_i = \sum_{j=1}^{N_l} S_{ij}e^{2H_i}\delta(y_j^l, -1) + \frac{C}{2}\sum_{j=N_l+1}^{N_e} S_{ij}e^{H_i-H_j} \tag{9}$$

and $\delta(x, y) = 1$ when $x = y$ and 0 otherwise. According to [16], F is minimized when we choose $h\left(\mathcal{X}\right)$ such that the unlabeled samples with maximum values of $|p_i - q_i|$ are classified by $h_i = \text{sign}(p_i - q_i)$. The optimal α that minimizes F is

$$\alpha = \frac{1}{4}\ln\frac{\sum_{i=N_l+1}^{Ne} p_i\delta(h_i, 1) + \sum_{i=N_l+1}^{Ne} q_i\delta(h_i, -1)}{\sum_{i=N_l+1}^{Ne} p_i\delta(h_i, -1) + \sum_{i=N_l+1}^{Ne} q_i\delta(h_i, 1)}. \tag{10}$$

That is to say, at each iteration, we can sample these unlabeled samples according to the weights $|p_i - q_i|$, label them with $z_i = 2 \cdot \text{sign}(p_i - q_i)$, and combine them with the labeled samples (labeled with $\mathbf{y}_l$) to learn a new discriminative embedding space and a new tensor classifier model $h\left(\mathcal{X}\right)$, obtained as shown in Figure 1. We detail the procedure of the semi-supervised improvement in Algorithm 2.

Discussion From Eq. (8) and Eq. (9), we can see that, if an unlabeled sample $\mathcal{X}_i$ is highly similar to the positive samples, but wrongly predicted to have the negative label by the un-improved tensor classifier, then this unlabeled sample has a large p_i and a small q_i. We sample this unlabeled sample and label it with $z_i = 2 \cdot \text{sign}(p_i - q_i)$. Likewise, we sample the top few most "mispredicted" samples for improving the original tensor classifier. All these sampled unlabeled samples help to compensate for the loss of discriminant information encoded in the embedding space of the un-improved tensor classifier and to encode most of the discriminant information. Grabner *et al*. [6] use similar mechanism to weight the unlabeled samples and hence use them to train their feature selection based boosting classifier. Their method uses all the unlabeled samples for training without selection. A direct comparison of results obtained from their method and ours is given in Section 3.2.

3. Experimental Results and Analysis

In this section, we present experimental results that validate the superior properties of our proposed tensor representation based discriminant tracker with semi-supervised improvement (SSI-TDT). As described in Section 1, we make two novel assumptions for our tracking approach, which are image-as-matrix assumption and semi-supervised improving assumption. With different assumptions, different object trackers are obtained. When we adopt the image-as-vector assumption, it is a vector representation based discriminant tracker with semi-supervised improvement (SSI-VDT). Our tracking approach without semi-supervised improving assumption results in TDT. TDT with margin improving assumption (*e.g*. ASSEMBLE) results in MI-TDT. We compare SSI-TDT with SSI-VDT, TDT, and MI-TDT to demonstrate the effectiveness of our assumptions. Additionally, we conduct a thorough comparison between SSI-TDT and eight recent and state-of-the-art visual trackers denoted as: IRTSA [13], Frag [1], VTD [8], VTS [9], MIL [2], IVT [21], and APG-ℓ_1 [3], SSOBT [6]. We implement these trackers using publicly available source codes or binaries provided by the authors. For fair evaluation, each tracker is run with appropriately adjusted parameters.

Implementation details All our experiments are done using MATLAB R2008b on a 2.83GHz Intel Core2 Quad PC with 4GB RAM. As shown in Figure 1, we use the particle filter in [21] to draw unlabeled samples from frame I_{t+1}, where we simply consider the object state information in 2D translation and scaling, and set the number N_u of particles to 600. We draw positive samples based on the tracking results at previous frames, and draw negative samples from the current frame I_t using the dense sampling method. In practice, we set the number of positive samples $n_{+1} = 50$, the number of negative samples $n_{-1} = 100$. We sample top 10% of the unlabeled samples for improvement of original classifier. This indicates that $N_l = 150$, and $N = 210$. All the image regions corresponding to these samples are normalized to templates of size 32×32, which indicates that $m_1 = m_2 = 32$. The column number of $\mathbf{U}$ and $\mathbf{V}$ in Algorithm 1 are both set to $l_1 = l_2 = 2$. The parameter $\hat{T}$ in Algorithm 2 is set to 2 for the sake of compromise between tracking speed and accuracy. The above parameter settings remain the same in all the experiments.

To evaluate SSI-TDT, we compile a set of 12 challenging video sequences consisting of 8-bit grayscale images. These videos include challenging appearance variations due to the changes in pose, illumination, scale, and the presence of occlusion and cluttered background. For quantitative comparison, two evaluation criteria are introduced, namely, center location error (CLE) and VOC overlap ratio (VOR) between the predicted bounding box B_p and ground truth B_{gt} (manually labeled) such that VOR $= \frac{\text{area}(B_p \bigcap B_{gt})}{\text{area}(B_p \bigcup B_{gt})}$. If VOR is larger than 0.5, then the target is considered to be successfully tracked. This can be used to evaluate the success rate (i.e., $\frac{\#successframes}{\#totalframes}$) of any tracking approach.

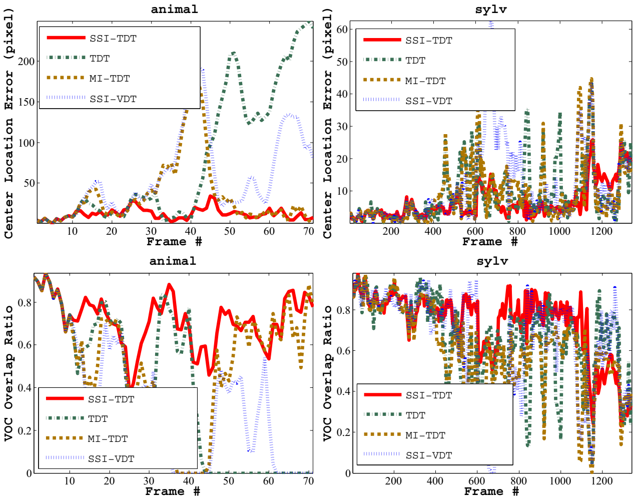


Figure 2: Quantitative comparison of the proposed tracker with different assumptions on two videos. The left two subfigures are associated with the tracking performance in CLE and VOR on the *animal* video, respectively; the right two subfigures correspond to the tracking performance in CLE and VOR on the *sylv* video, respectively.

	animal	*boat*	*coke11*	*david*	*dudek*	*football*	*sylv*	*woman*
SSI-TDT	**0.94**	**0.81**	0.80	**0.99**	**1.00**	**0.73**	**0.88**	**0.87**
TDT	0.44	0.27	**0.83**	0.29	0.58	0.60	0.79	0.59
MI-TDT	0.59	0.59	0.78	0.34	0.93	0.62	0.73	0.77
SSI-VDT	0.27	0.62	0.10	0.48	0.78	0.29	0.72	0.09

Table 1: Quantitative comparison of the proposed tracker with different assumptions on eight videos. The table reports their tracking success rates.

3.1. The effectiveness of our approach

As previously claimed, 2^{nd}-order tensor (image-as-matrix) representation methods can retain more useful information than image-as-vector representation methods, and the semi-supervised improvement technique can enhance the tensor based discriminant tracker much more than margin improving techniques. To demonstrate the effectiveness of our two assumptions, we design several experiments on eight videos. Figure 2 quantitatively shows some experimental results of the proposed tracker with different assumptions on two of the eight videos. Table 1 reports their tracking success rates on the eight videos. From Figure 2 and Table 1, we can see that semi-supervised improvement (SSI) technique always enhances TDT, and the enhancement is always much more notable than margin improving

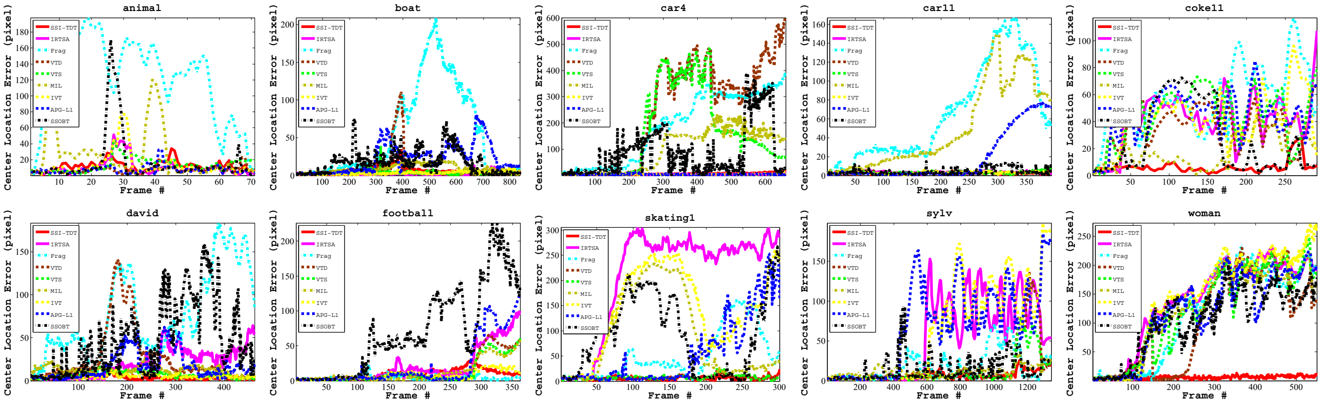


Figure 3: Quantitative comparison of different trackers in terms of CLE on ten videos.

(MI) technique's, except for the *coke11* video. The reason is the *coke11* video has few challenges in illumination, scale changes and background clutter, while the challenges caused by the presence of occlusion and varying viewpoints are obviously noted. TDT can handle the later challenges well, so SSI technique and MI technique are not required in the *coke11* video. TDT can't handle the challenges in large scale variation (*e.g. david*, *dudek*), background clutter (*e.g. boat*, *football*), fast motion and blur (*e.g. animal*) well. In addition, although the SSI technique and discriminant information are used, the image-as-vector representation based tracker (SSI-VDT) still can't meet the challenges caused by the presence of occlusion (*e.g. coke11*, *football*, *woman*), fast motion and blur (*e.g. animal*) well because of the limitations of image-as-vector representation.

3.2. Comparison with competing trackers

To show the superiority of SSI-TDT over other competing trackers, we perform experiments using IRTSA [13], Frag [1], VTD [8], VTS [9], MIL [2], IVT [21], APG-ℓ_1 [3] and SSOBT [6] on twelve videos. Figure 3 shows the center location errors of the nine evaluated trackers on ten videos. Table 2 and Table 3 summarize the average center location errors in pixels and the success rates over all the twelve videos, respectively. Over all, the proposed tracking approach performs favorably against state-of-the-art trackers. Figure 4(l) demonstrates that the proposed approach outperforms the other approaches significantly when heavy occlusion and pose variation appear simultaneously, as in the *coke11*, *skating1* videos. The incremental subspace learning based approaches (*e.g.* IRTSA, IVT) are able to capture appearance variations due to illumination change (*e.g. car4*, *david*), scale change (*e.g. car4*, *david*, *dudek*), and background clutter (*e.g. car11*, *dollar*), but they fail in the more challenging videos which include drastic pose variation and heavy occlusion. VTD and VTS achieve good results over most of the videos due to the combination of a set of basic observation (or motion) models, but they achieve higher tracking errors and lower success rates than our approach.

	IRTSA	Frag	VTD	VTS	MIL	IVT	APG-ℓ_1	SSOBT	SSI-TDT
animal	9.8	116.5	**7.0**	13.4	24.0	12.7	**7.2**	16.4	11.7
boat	**2.6**	54.4	8.4	3.5	9.0	**3.1**	22.3	17.0	5.1
car4	4.8	185.6	245.2	160.0	97.3	**4.6**	**4.2**	88.0	6.7
car11	3.2	65.5	3.0	3.0	50.0	**2.8**	18.1	4.1	**2.5**
coke11	41.2	58.2	35.4	45.1	**18.4**	41.8	44.5	26.2	**6.3**
david	20.3	77.2	26.5	7.0	12.5	**4.7**	17.2	46.9	**4.1**
dollar	11.9	55.7	5.1	**4.6**	5.9	17.5	64.1	67.9	**4.7**
dudek	**7.7**	94.2	10.3	70.5	19.7	9.3	161.4	52.0	**8.4**
football	20.8	**4.1**	13.4	12.0	12.6	**7.7**	22.3	70.6	9.3
skating1	210.2	50.3	**6.7**	7.7	91.5	141.2	53.8	92.1	**5.9**
sylv	53.3	17.6	15.7	11.4	**11.2**	62.5	68.6	14.2	**6.7**
woman	138.9	117.9	**102.8**	122.3	121.6	144.5	118.8	104.9	**5.7**

Table 2: Average center location error (in pixels). The best and second best results are shown in red and blue fonts.

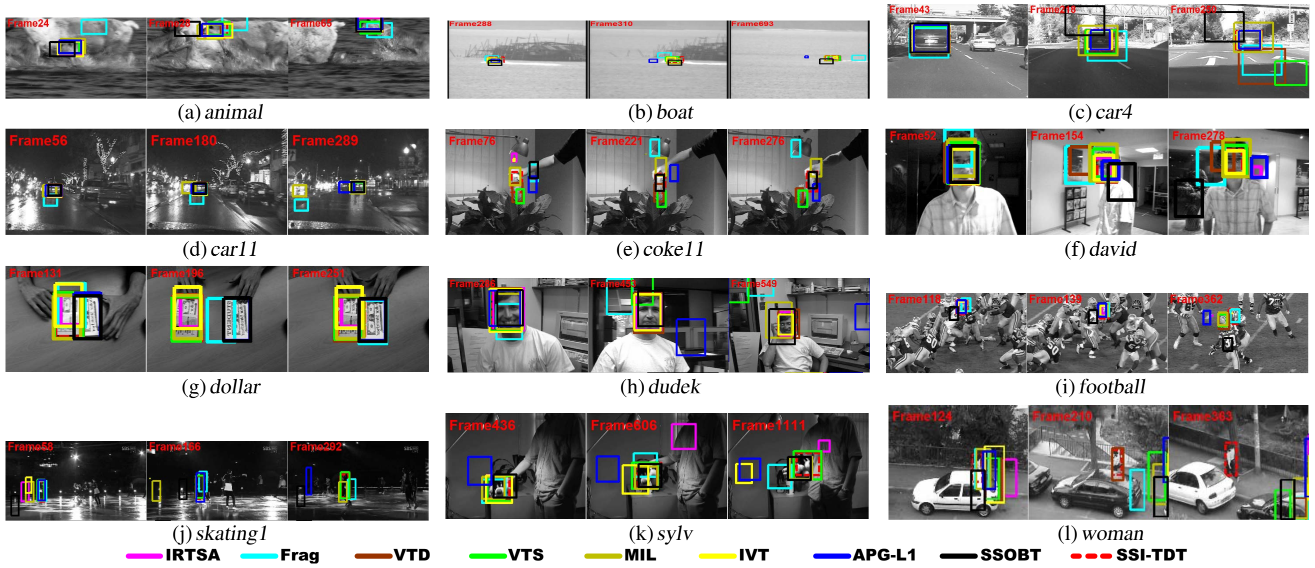
	IRTSA	Frag	VTD	VTS	MIL	IVT	APG-ℓ_1	SSOBT	SSI-TDT
animal	0.93	0.10	**0.97**	0.87	0.75	0.89	**0.97**	0.89	**0.94**
boat	0.10	0.35	0.51	**0.62**	0.54	0.14	0.13	0.35	**0.81**
car4	**1.00**	0.29	**0.35**	**0.35**	0.29	**1.00**	**1.00**	0.30	**1.00**
car11	**1.00**	0.10	**0.96**	0.95	0.15	**1.00**	0.68	0.79	**1.00**
coke11	0.14	0.05	0.07	0.07	0.32	0.10	0.08	**0.37**	**0.80**
david	0.34	0.08	0.61	**0.94**	0.79	0.90	0.43	0.32	**0.99**
dollar	**0.45**	0.41	**1.00**	**1.00**	**1.00**	**1.00**	0.39	0.33	**1.00**
dudek	**1.00**	0.53	**1.00**	0.79	**0.82**	**1.00**	0.52	0.68	**1.00**
football	0.36	**0.97**	**0.78**	**0.78**	0.75	0.41	0.76	0.31	0.73
skating1	0.13	0.33	**0.90**	**0.90**	0.26	0.11	0.51	0.26	**0.96**
sylv	0.43	0.62	0.81	**0.83**	0.75	0.45	0.32	0.67	**0.88**
woman	0.16	0.26	**0.31**	0.19	0.21	0.20	0.22	0.19	**0.87**

Table 3: Success rate of tracking approaches. The best and second best results are shown in red and blue fonts.

Figure 4(g) shows that Frag, APG-ℓ_1 and SSOBT are easily confused by the impostor object.

4. Conclusion

In this paper, we have proposed an effective and robust discriminant tracking approach with semi-supervised improvement based on 2^{nd}-order tensor representation. The superiority of our approach can be attributed to: 1) the specially designed two graphs for modeling the local geometrical and discriminative structure of the 2^{nd}-order tensor samples; 2) the introduced semi-supervised improvement technique for compensating for the loss of discriminant information. This 2^{nd}-order tensor representation based approach retains more useful information than image-as-vector representation based one, which makes the proposed tracker able to address the challenges caused by heavy occlusion and pose variation. Experimental results compared


(a) *animal* (b) *boat* (c) *car4*

(d) *car11* (e) *coke11* (f) *david*

(g) *dollar* (h) *dudek* (i) *football*

(j) *skating1* (k) *sylv* (l) *woman*

Figure 4: Screenshots of some sampled tracking results of evaluated approaches on twelve challenging videos.

with several state-of-the-art trackers on challenging videos demonstrate the effectiveness and robustness of the proposed approach.

Acknowledgment This work is partly supported by NSFC (Grant No. 60935002), the National 863 High-Tech R&D Program of China (Grant No. 2012AA012504), the Natural Science Foundation of Beijing (Grant No. 4121003), and The Project Supported by Guangdong Natural Science Foundation (Grant No. S2012020011081).

References

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

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

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust ℓ_1 tracker using accelerated proximal gradient approach. *In CVPR*, 2012.

[4] K. P. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. *In ACM SIGKDD*, 2002.

[5] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. *In BMVC*, 2006.

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

[7] X. He, D. Cai, and P. Niyogi. Tensor subspace analysis. *In NIPS*, 2005.

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

[9] J. Kwon and K. Lee. Tracking by sampling trackers. *In ICCV*, 2011.

[10] G. Li, L. Qin, Q. Huang, J. Pang, and S. Jiang. Treat samples differently: object tracking with semi-supervised online covboost. *In ICCV*, 2011.

[11] P. Li and Q. Sun. Tensor-based covariance matrices for object tracking. *In ICCV Workshops*, 2011.

[12] X. Li, A. Dick, H. Wang, C. Shen, and A. van den Hengel. Graph mode-based contextual kernels for robust svm tracking. *In ICCV*, 2011.

[13] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Robust visual tracking based on incremental tensor subspace learning. *In ICCV*, 2007.

[14] X. Li, W. Hu, Z. Zhang, M. Zhu, and J. Cheng. Visual tracking via incremental log-euclidean riemannian subspace learning. *In CVPR*, 2008.

[15] X. Li, C. Shen, Q. Shi, A. Dick, and A. van den Hengel. Non-sparse linear representations for visual tracking with online reservoir metric learning. *In CVPR*, 2012.

[16] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *Trans. on PAMI*, 31:2000–2014, 2009.

[17] X. Mei and H. Ling. Robust visual tracking using ℓ_1 minimization. *In ICCV*, 2009.

[18] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient ℓ_1 tracker with occlusion detection. *In CVPR*, 2011.

[19] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. *In CVPR*, 2006.

[20] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. *In Workshop on Applications of Computer Vision*, 2005.

[21] D. A. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *Int. J. Comp. Vis.*, 77(1):125–141, 2008.

[22] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Machine Learning Research*, 8, 2007.

[23] M. A. O. Vasilescu and D. Terzopoulos. Tensor textures: multilinear image-based rendering. *In ACM SIGGRAPH*, 2004.

[24] Q. Wang, F. Chen, and W. Xu. Tracking by third-order tensor representation. *Trans. on SMCB*, 41(2):385–396, 2011.

[25] J. Wen, X. Gao, X. Li, and D. Tao. Incremental learning of weighted tensor subspace for visual tracking. *In Int. Conf. on Sys., Man, and Cyb.*, 2009.

[26] Y. Wu, J. Cheng, J. Wang, and H. Lu. Real-time visual tracking via incremental covariance tensor learning. *In ICCV*, 2009.

[27] S. Yan, D. Xu, S. Lin, T. S. Huang, and S. F. Chang. Element rearrangement for tensor-based subspace learning. *In CVPR*, 2007.

[28] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *Trans. on PAMI*, 29:40–51, 2007.

[29] J. Ye. Generalized low rank approximations of matrices. *Machine Learning*, 61:167–191, 2005.

[30] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. *In NIPS*, 2004.

[31] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

[32] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *In NIPS*, 2005.

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

[34] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. *In CVPR*, 2012.

[35] X. Zhang, W. Hu, S. Maybank, and X. Li. Graph based discriminative learning for robust and efficient object tracking. *In ICCV*, 2007.