Local Subspace Collaborative Tracking

Lin Ma[1,3], Xiaoqin Zhang[2], Weiming Hu[1], Junliang Xing[1], Jiwen Lu[3] and Jie Zhou[3]

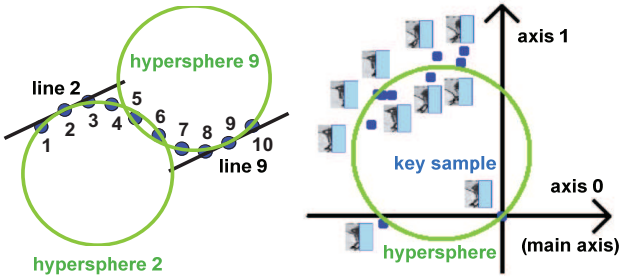
1. NLPR, Institute of Automation, Chinese Academy of Sciences, China
2. College of Mathematics and Information Science, Wenzhou University, China
3. Department of Automation, Tsinghua University, China

Abstract

Subspace models have been widely used for appearance based object tracking. Most existing subspace based trackers employ a linear subspace to represent object appearances, which are not accurate enough to model large variations of objects. To address this, this paper presents a local subspace collaborative tracking method for robust visual tracking, where multiple linear and nonlinear subspaces are learned to better model the nonlinear relationship of object appearances. First, we retain a set of key samples and compute a set of local subspaces for each key sample. Then, we construct a hypersphere to represent the local nonlinear subspace for each key sample. The hypersphere of one key sample passes the local key samples and also is tangent to the local linear subspace of the specific key sample. In this way, we are able to represent the nonlinear distribution of the key samples and also approximate the local linear subspace near the specific key sample, so that local distributions of the samples can be represented more accurately. Experimental results on challenging video sequences demonstrate the effectiveness of our method.

1. Introduction

Visual object tracking plays an important role in many vision applications such as video surveillance and motion analysis. To build up robust trackers, many feature representation methods have been proposed to model the appearance of objects [4, 3, 31, 19, 10]. Generally, the appearance of objects is complex and may present different distributions. Subspace analysis is an effective technique to model object appearance and has been successfully applied to visual tracking. Principal component analysis (PCA) is one representative subspace method, which assumes samples distribute linearly and aims to learn a low-dimensional subspace by maximizing the variance of samples [22]. In visual tracking, objects usually vary across time and PCA cannot be directly employed to model object appearance.



(a) (b)

Figure 1. **hypersphere.** (a) hypersphere (green circles) of 2D (blue) points. Line 2 and Line 9 passing points 2 and 9 are local linear subspaces about point 2 and 9 respectively. The hyper spheres are tangent to the local linear subspaces (1D). As the figure shows, the hypersphere is able to represent the local point distributions more accurately than the local linear subspace. (b) Local nonlinear subspace of a key sample (at the origin). The object is divided into four object parts, i.e. left, right, bottom and top. Here, we show the key samples of the left object part. For clarity, we show the 1D sphere (green circle) in 2D linear subspace (the two axes). The blue points represent the object appearances. The nonlinear distribution of the key samples is approximated with the hypersphere. The sphere passes the key samples and is tangent to the main axis (axis 0) approximately. As the figure shows, when the object experiences pose variation, the samples are distributed on a nonlinear hypersphere approximately. Thus, with hypersphere, we can represent the object appearance distribution more accurately. Also, by making the hypersphere tangent to the corresponding axis, the local linear subspace can be approximated by the hypersphere.

To address this, the incremental PCA method is proposed to update the linear subspace incrementally so that the model can well adapt to the varying appearance during tracking [17].

Linear subspace is less likely to over-fit samples, but easily fails to represent the real sample distribution. To resolve the nonlinear distribution problem, kernel based methods have been proposed [21, 2]. While nonlinearities can be ad-

dressed, high computational cost is usually required. Even if the computational cost can be reduced by computing the inner product with the kernel function, the information containing in the Euclidean distances between samples is lost. Moreover, if an unsuitable kernel is used, the performance is worse. Manifold methods supply another way to resolve the nonlinear problem [20, 18]. For example, Li *et al.* [15] assume that samples are distributed on a hypersphere and can be warped to a linear tangent subspace to perform linear operations by using the logarithm and exponent operations. With the Riemannian subspace, an effective representation of the object appearances can be obtained [15]. However, the warping process is not explicit, especially in the high dimensional space. Moreover, the original Euclidean distances between samples cannot be well preserved.

In this paper, we propose a new local subspace collaborative tracking method to model the nonlinear distribution of samples and preserve the similarity of samples. Figure 1 shows the basic idea of our proposed approach. In our model, we use multiple linear and non-linear subspaces to model the nonlinear relationship of object appearances. Specifically, we retain a set of key (representative) samples and compute a local nonlinear subspace for each key sample to represent the local distribution. To represent the local nonlinear distribution of each frame, the sphere is constructed to pass the local samples. On the other hand, a local linear subspace for each key frame is also computed because it is more robust to noises. Therefore, we first obtain a local linear subspace for each key sample, and then form the sphere in the linear subspace. These two types of subspaces are both effective in representing the object appearance, thus we take advantage of them and propose a local subspace collaborative appearance model. When the object experiences pose variation, as shown in Figure 1(b), the pixels change positions on the object subimage. And then the samples are distributed nonlinearly. Our model represents the nonlinear distribution effectively. Moreover, the linear subspace reserves the majority of Euclidean distances between samples. Then the hypersphere is also able to reserve the Euclidean distances between samples approximately as the sphere does not revise the Euclidean distances between samples in the linear subspace. Figure 2 shows the flowchart of our approach.

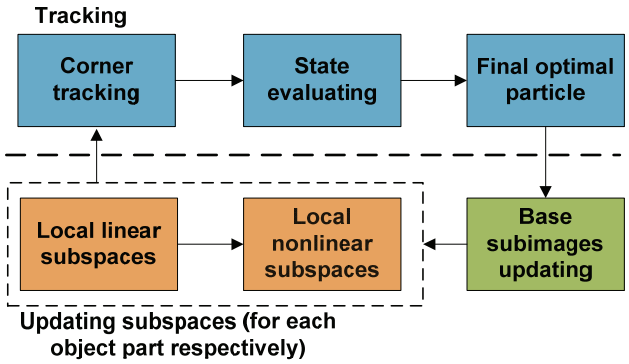


Figure 2. **Flowchart of our system.** The system contains two main parts: Tracking based on the local linear and nonlinear subspaces collaborative model and updating the appearance model. When performing tracking, we first obtain an initial state with corner tracking, and then sample a set of candidate states around the initial state. The state with largest evaluation value is selected as the final object state. When updating the appearance model, for each key sample, we first update the local linear subspace, and then update the corresponding local nonlinear subspace. The four object parts, i.e. left, right, bottom and top, are dealt with separately during tracking (the evaluations of the four parts are combined) and model updating.

2. Related Work

Searching Principle and Appearance Model: When performing object tracking, there are two important components to consider, i.e. searching principle and appearance model. Meanshift and particle filter are two widely used searching principles. Based on meanshift, Bradski [4] proposed the Camshift (Continuously Adaptive Mean-Shift) method to track faces using color histogram. Meanshift is easily trapped in local optimal solutions. In contrast, particle filter obtains more global solution by utilizing sampling methods [9, 16, 10]. Appearance model is also very important, and good appearance model is able to tackle various challenges, such as drastic pose variation and severe occlusion [28, 25, 23, 11, 6, 8]. PCA subspace is a widely used appearance model which represents the linear sample distributions [17]. Li *et al.* [14] projected the samples on Riemannian manifold to a tangent subspace and computed the PCA subspace in the new tangent subspace. Sparse representation is another widely used object tracking approach, which reconstructs the object sample with only a few dictionary templates [10, 24, 26]. Mei and Ling [16] introduced sparse representation into visual tracking, and formed the dictionary with both positive templates and negative templates. Wang [24] learned non-negative dictionary for robust visual tracking. To represent the object structure and tackle occlusion and pose variation problem, part-based models are utilized by many researchers [19, 14, 19, 28]. Adam *et. al.* [1] divided the object appearance into a set of fragments, and evaluated the candidate state based on comparing the similarity between fragments. In this way, the occlusion problem can be effectively tackled. Xu *et. al.* [10] formed a dictionary with object patches and obtain the best object state through alignment-pooling of sparse coefficients. Recently deep learning is also utilized to perform object tracking. With convolution neural networks, Li *et. al.* obtained discriminative features for visual tracking [13].

Nonlinear Appearance Model: Many appearance mod-

els assume that the samples are distributed linearly. However, in many conditions the object appearances are distributed nonlinearly. To obtain accurate low dimensional nonlinear subspace, various manifold learning methods have been proposed, where local relations of samples are prescored in the learned feature subspace [20, 18, 32, 27]. For example, Zhang *et al.* [29] proposed a robust nonnegative graph embedding (RNGE) method. RNGE tackles the noise and graph unreliability problems robustly with the joint sparsity in both graph embedding and reconstruction. Tiwari *et al.* [21] used the kernel method in graph embedding to represent the nonlinear distribution of object appearance. Some researchers evaluated the similarities between samples with Non-Euclidean distance to tackle the nonlinear sample distribution problem. For example, Adamn *et al.* [1] computed the distances between features of patches with earth mover's distance which is non-Euclidean distance. Riemannian subspace which represents the nonlinear distribution of samples by assuming the samples are distributed on a hypersphere, is also an important appearance model [15]. To represent the part information, Li *et al.* [14] extended the method in [15] by dividing the object appearance into patches and proposed a new Riemannian subspace based method, SLAM (Spatial Log-Euclidean Appearance Model). SLAM warps each patch to linear tangent subspace and forms the object appearance model with the warped samples. Compared with the Riemannian subspace, our method represents the nonlinear sample distribution more explicitly and also reserves more Euclidean distances between samples. Moreover, our method also approximates the local linear subspaces by making the hypersphere tangent to the local linear subspace.

3. System Overview

Our aim is to obtain the object state at each frame. Let X_t denote the object state at frame t. The object state represents the object position and object height and width scales. In each frame, we first perform corner tracking and obtain an initial state. Then we sample N_S candidate states around the initial state according to Gaussian distribution. Each candidate state is evaluated with the proposed linear and nonlinear subspaces collaborative model. The candidate state with the best evaluation is selected as the optimal state. The object state at frame $t-1$ is defined as the beginning state at frame t.

To perform corner tracking, K representative object subimages (the image within the object bounding box) are retained as base subimages B_i, $i = 1, 2, ..., K$. When performing tracking at frame t, the base subimage with the smallest Euclidean distance to object appearance at frame $t-1$ is selected as the current base subimage $\widetilde{B}$. We detect Harris corners on $\widetilde{B}$, and compute the translation of each corner separately. For the corner P_i, we search the point P_i^c with the most similar texture within a neighborhood, and then obtain the translation of P_i. Let E_T be the l_1 norm distance between two texture features. Then the texture similarity is computed as $\exp(-E_T)$. The object translation is defined as the average of the translations of the corners where the weight of each corner is proportional to the texture similarity. The object size keeps invariant when performing corner tracking. With corner translation, we obtain the coarse object state, and compute the fine state with the proposed model.

4. Local Subspace Collaborative Tracking

We evaluate the candidate object state with the proposed appearance model. During tracking, we warp the object appearance specified by the candidate object state to a normalized 32×32 (gray) subimage. To tackle occlusions, the object subimage is divided into four parts (i.e. left, right, bottom and top). These four parts are evaluated separately, and the sum of their evaluations is defined as the evaluation of the candidate state. Figure 1(b) shows the left part of the sample. Each part of the candidate sample is unfolded to a vector. For each part, we use the corresponding areas of the base subimages to form K key samples (Figure 3).

For a given part of the candidate sample, assume I be the unfold vector at time t, and N_D be the feature dimension. Let the key samples be S_k, where $k = 1, ..., K$. We compute the local linear subspace and local nonlinear subspace for each key sample, respectively. Local subspaces represent the samples' local distributions. Generally, samples are distributed on some kinds of geometries. Compared with other geometries such as hyper ellipse, hypersphere has relatively fewer parameters, and is easier to be constructed and less likely to be overfitted. Moreover, hypersphere can represent the nonlinear sample distribution, and obtain more accurate distance between candidate sample. Thus, we use the hypersphere to represent the local nonlinear subspace. As Figure 1 shows, the hypersphere can represent the samples's nonlinear distribution effectively when pose variation occurs.

Generally, the feature dimension of the sample is larger than the number of key samples. If we directly compute the hypersphere based on these key samples, we cannot obtain a unique solution. Thus, for one key sample (we take S_k as an example in this section), we first project all the key samples to the local linear subspace of the specific key sample, and then construct the hypersphere in the new subspace.

4.1. Local Linear Subspaces

Linear subspace is an important model in representing the object appearance [17]. For S_k, we obtain the local linear subspace U_k with the eigenvectors about $\mathcal{A} = [w_1(S_1 - S_k), \cdots, w_K(S_K - S_k)]$, where w_i is S_i's, $i = 1, ..., K$ weight about S_k (Figure 3). Compared with the

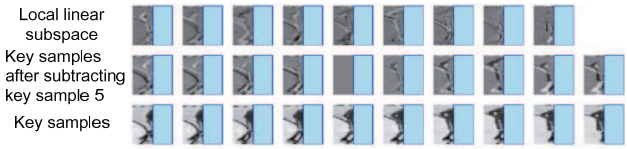


Figure 3. **Local linear subspace of key sample 5.** Each key sample subtracts key sample 5 and obtains a new sample (middle row). The local linear subspace of key sample 5 is obtained based on the new samples (top row). The matrices are warped to [0, 255] for display.

conventional linear subspace, S_k is not the mean of the samples, and U_k represents the samples' distributions which is not relative to the mean sample. Let $\widetilde{D}_i$ be the Euclidean distance between S_i and S_k. We define

$$w_i \propto \exp\{-\widetilde{D}_i\}, \tag{1}$$

where U_k is updated adaptively when new key samples come. Specifically, when K new key samples are obtained, the subspace of the K new samples for S_k is defined as the new linear subspace U_k. The dimension of U_k is defined as N_u.

4.2. Local Nonlinear Subspaces

We construct the hypersphere H_k, i.e. the local nonlinear subspace, for S_k in U_k (Figure 4). By making the linear subspace and nonlinear subspace work collaboratively, more accurate representation of the object sample distribution is obtained. To obtain the unique solution to H_k, we first project all key samples into U_k, and then obtain H_k in U_k. In U_k, we define that S_k and U_k correspond to the origin $\mathbf{0}$ and the new axis $\widetilde{I}_l$ respectively, where $\mathbf{0}$ is a vector where each clement is 0, and its size is N_u, $\widetilde{I}_l$ is an identity matrix of dimension $N_u \times N_u$. Here each column vector of $\widetilde{I}_l$ is considered as an axis. Let $\overline{S}_i$, $i = 1, ..., K$ be the projection of S_i in U_k, and $\overline{U}$ be the projection of U' (the first $N_u - 1$ dimensions of U_k) in U_k. We obtain

$$\overline{S}_i = U_k^T(S_i - S_k), \tag{2}$$

$$\overline{U} = U_k^T U'. \tag{3}$$

With the projected key samples, we construct H_k. Let D_k be the center of sphere H_k and r_k be the radius of H_k. Then the sphere is defined as

$$||x - D_k||_2 = r_k, \tag{4}$$

where x is a sample vector in U_k. To represent the local distribution of the (projected) key samples, H_k needs to pass the key samples, which is

$$||\overline{S}_i - D_k||_2 = r_k, i = 1, ..., K. \tag{5}$$

Generally, the linear subspace is robust to noises, especially when the considered sample is very near to the key sample. Thus we make H_k tangent to $\overline{U}$ to approximate $\overline{U}$, that is,

$$\overline{U}^T(\overline{S}_k - D_k) = \overline{U}^T D_k = \mathbf{0}, \tag{6}$$

where $\overline{S}_k$=**0** according to (2). If we consider $\overline{U}$ as a hyper plane, with (6) $\overline{S}_k - D_k$ can be considered as $\overline{U}$'s norm (Figure 4(b)). In U_k, the sphere H_k is of dimension $N_u - 1$, so H_k is only able to approximate the plane of dimension $N_u - 1$ at most. Thus, we define $\overline{U}$'s dimension as $N_u - 1$. Under the constraints of (5) and (6), we obtain

$$D_k = L_0^{-1} L_1, \tag{7}$$

where

$$L_0 = 4\sum_{i,j} w_i w_j (\overline{S}_i - \overline{S}_j)(\overline{S}_i - \overline{S}_j)^T + \lambda \overline{U}\overline{U}^T, \tag{8}$$

$$L_1 = \sum_{i,j} 2 w_i w_j (\overline{S}_i - \overline{S}_j)(\overline{S}_i^T \overline{S}_i - \overline{S}_j^T \overline{S}_j), \tag{9}$$

and λ is a constant to tune the constraints' importance of (5) and (6) in (7). The computation process for (7) is shown in Appendix A. With D_k, we obtain

$$r_k = ||\overline{S}_k - D_k||_2 = ||D_k||_2. \tag{10}$$

We only consider $\overline{S}_k$ in (10) to reduce the noise influence of other key samples.

Let O_t be the object appearance at frame t. To evaluate the rth object part I of the current sample, we break the distance between I and S_k's subspace into two parts, the distance E^u between I and U_k, and the distance E^h between H_k and the projection of I in U_k. We define

$$E^u = ||(I - S_k) - U_k U_k^T (I - S_k)||_2, \tag{11}$$

$$E^h = |r_k - ||U_k^T (I - S_k) - D_k||_2|, \tag{12}$$

and then define

$$p_k^r(O_t|X_t) \propto \exp\{-(E^u + E^h)\} \tag{13}$$

as the evaluation of I corresponding to key sample k. Then, with the key sample k^* having smallest Euclidean distance to I, i.e. $k^* = \arg\min_k ||I - S_k||_2$, the evaluation of I is defined as

$$p^r(O_t|X_t) = p_{k^*}^r(O_t|X_t). \tag{14}$$

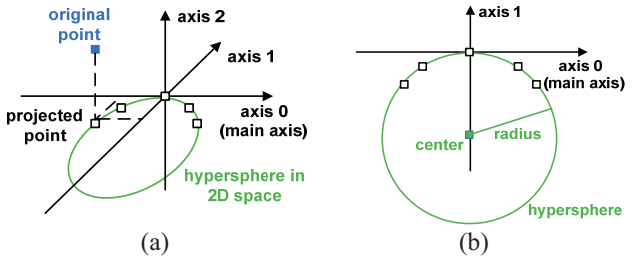
We obtain the likelihood $p(O_t|X_t)$ of X_t by combining the evaluations of the four part samples and obtain

$$p(O_t|X_t) = \sum_r p^r(O_t|X_t). \tag{15}$$

Algorithm 1: Constructing the local subspaces.

Input: $\mathcal{S}_k, k = 1, ..., K$.
Output: $U_k, H_k, k = 1, ..., K$.

```
//Compute local linear subspaces.
for k = 1, ..., K do
    Compute w_i, i = 1, ..., K with (1).
    Construct A.
    Obtain U_k by performing PCA on A.
end
//Compute local nonlinear subspaces.
for k = 1, ..., K do
    Compute D_k with (7).
    Compute r_k with (10).
end
```



(a) (b)

Figure 4. **Local nonlinear subspace.** To simplify the example, the original point is 3D and the projected point is 2D. (a) Original space. (b) Nonlinear subspace of one key sample (at the origin). We first project the key samples to the local linear subspace of one key sample, and then construct the hypersphere in the new linear subspace.

4.3. Updating Base Subimages

We retain the object samples of the initial K frames as the initial base subimages. To adapt to the appearance variation, we update the base subimages with the saved subimages and update the local linear and nonlinear subspaces based on the new base subimages. Let $\overline{E}_t$ be the average of all frames' evaluations (15) until frame t. Then if the evaluation E_t of the optimal sample I_t is larger than a constant scale of $\overline{E}_t$, the object appearance normally has no large variation and then we consider I_t is not polluted, e.g. occluded, and save I_t, or else drop I_t. For accuracy, only not polluted samples are used to update the subspaces.

We update the base subimages every M saved object subimages. We form a set of $K + M$ candidate subimages with previous K base subimages and the new M saved subimages. Then we select K candidate subimages to form the new base subimages. For each candidate subimage, we compute the sum of the Euclidean distances between the current candidate subimage and other candidate subimages. The selected subimages correspond to the largest K sums. In this way, we can make the base subimages represent various object appearance forms. When computing the distances between candidate subimages, if the two subimages both are previous base subimages, the distance between them is multiplied by a constant s_1 ($s_1 = 1.5$ in this paper). Increasing s_1 increases the probability of selecting the old subimages, and vice verse. With the new base subimages, we obtain the key samples for each of the four object parts. Then, for a part, with the new key samples, we update the local subspaces as Algorithm 1. Algorithm 2 shows the procedure of our tracking system.

Algorithm 2: The tracking system.

Input: X_{t-1},F_t (frame image at frame t).
Output: X_t.

```
Perform corner tracking and obtain an initial state.
Sample a set of candidate states X_t^i, i = 1, ..., N_S
around the initial state.
Evaluate each candidate state with (15).
Update base subimages every M saved subimages,
and obtain the new key samples for each object part.
Update subspaces with the new key samples.
```

5. Experiments

In this section, we first present the implementation details of our method. Then, we investigate the performance of our approach where only the linear subspaces are used. Lastly, we compare our approach with eleven state-of-the-art trackers.

5.1. Implementation Details

We test our method on the 51 benchmark videos [25] which involve various challenges. The experiments are conducted on a PC with a 2.5 GHz Intel CPU with 8GB RAM. We set $N_S = 150$, K=10, $N_u = 9$, $\lambda = 0.02$, and update the system every 5 saved samples. The state in the first frame is manually set. The run time of our method is around 0.4 sec/frame. We compare our method with eleven state-of-the-art methods: IVT [17], Frag [1], VTD [12], ALSA [10], CT [30], SLAM [14], Struck [7], SCM [33], ColorT [5], GTPR [6] and KCF [8]. For the comparing methods, we utilize the source codes provided by the authors. We utilize the AUC values of pixel center error and PASC overlap evaluation to evaluate the tracking performance. The precision plot is plot over [0,50] with interval 10, and the success plot is plot over [0,1] with interval 0.2.

5.2. Effectiveness of Utilizing hypersphere

We compare our method with the method only using linear subspace (*SubS*) on ten video sequences to test the effectiveness of utilizing hypersphere. The ten video sequences are *Basketball*, *Couple*, *David3*, *FaceOcc2*, *Fleet-*

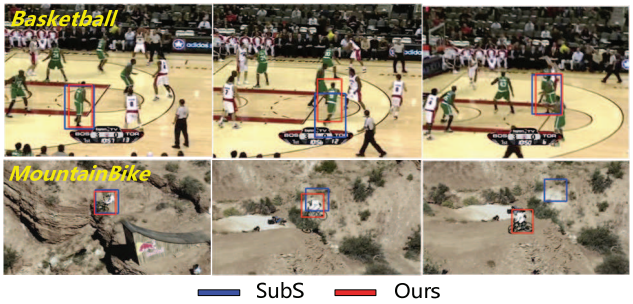


Figure 5. Sample frames of SubS and our method on two videos. The objects in both the two videos suffer from pose variation and the object samples are distributed nonlinearly. Our method obtains better performance than SubS due to the ability to represent nonlinear sample distribution.

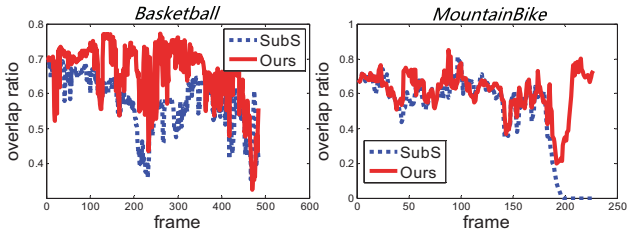


Figure 6. Overlap ratio at each frame of two video sequences. Our method represents the object state more accurately than *SubS* on the two videos.

Table 1. AUC values of two criteria of *SubS* and *Ours*.

	SubS	Ours
Center	0.671	**0.762**
PASC	0.575	**0.630**

Face, *MountainBike*, *Singer1*, *Subway*, *Jogging.1*, *Jogging.2*. Deformation or rotation occurs in these ten videos which makes the sample distribute nonlinearly to some extents. By utilizing hypersphere, our method is able to represent the nonlinear distribution of the object appearance, and then obtain more accurate distance between candidate sample and the sample distribution. Moreover, as the hypersphere is tangent to the local linear subspace, our method manages to approximate the local linear subspace. This makes the model less over-fit samples and robust to local clutters. The AUC values about pixel center error and PASC overlap are shown in Table 1. From Table 1, we see that on both the two criteria, our method achieves better performance. Figure 5 shows some sample frames of *SubS* and our method on *Basketball* and *MountainBike*. The objects change poses on the two videos, and the positions of the object pixels also vary which makes the object appearance change nonlinearly. With hypersphere, we obtain more accurate representation of object sample distribution and achieve better performances. Figure 6 shows the overlap ratio at each frame of *Basketball* and *MountainBike*. And Figure 7 shows the precision plot and the success plot of *SubS* and our method on the ten videos.

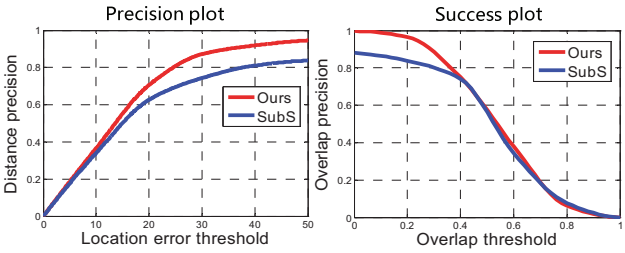


Figure 7. Precision plot and success plot of *SubS* and our method on ten videos. On both the two criteria, our method obtains larger AUC values than *SubS*.

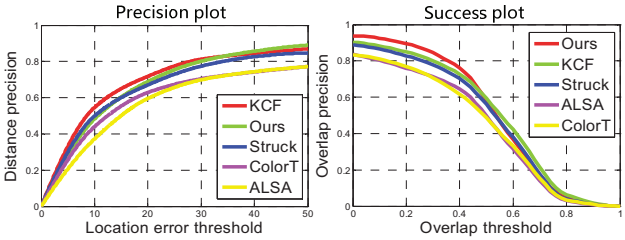


Figure 8. Precision plot and success plot of our method and four comparing methods, ColorT [5], ALSA [10], KCF [8] and Struck [7], on the 51 benchmark videos. The five methods are the top five according to Table 2 under the overlapping ratio. Our method obtains the largest success AUC value and second largest precision AUC value.

5.3. Comparison with State-of-the-arts

Quantitative Analysis: We compare our method with eleven state-of-the-art methods on the 51 benchmark video sequences. The AUC values of our method and the comparing methods according to two kinds of criteria are shown in Table 2. From the table, we see that our method achieves the best AUC value under the overlapping ratio criterion, and second best AUC value under the pixel center error criterion. Figure 8 shows the precision plot and the success plot of our method and four comparing methods on the 51 benchmark video sequences. The four methods are the best four comparing methods according to success AUC value. We also show the performance of each comparing method on the 11 attributes in Figure 9. From Figure 9, we see that under both the pixel center error criterion and the overlapping ratio criterion and on all the attributes, our method ranks among the top three. Especially, our method obtains the best performance on DEF attribute under both the two criteria, which demonstrates the effectiveness of our method in representing nonlinear sample distributions. Frag [1] uses a kind of non-Euclidean distance, Earth Movers distance, to evaluate the similarity between two histograms. But Frag

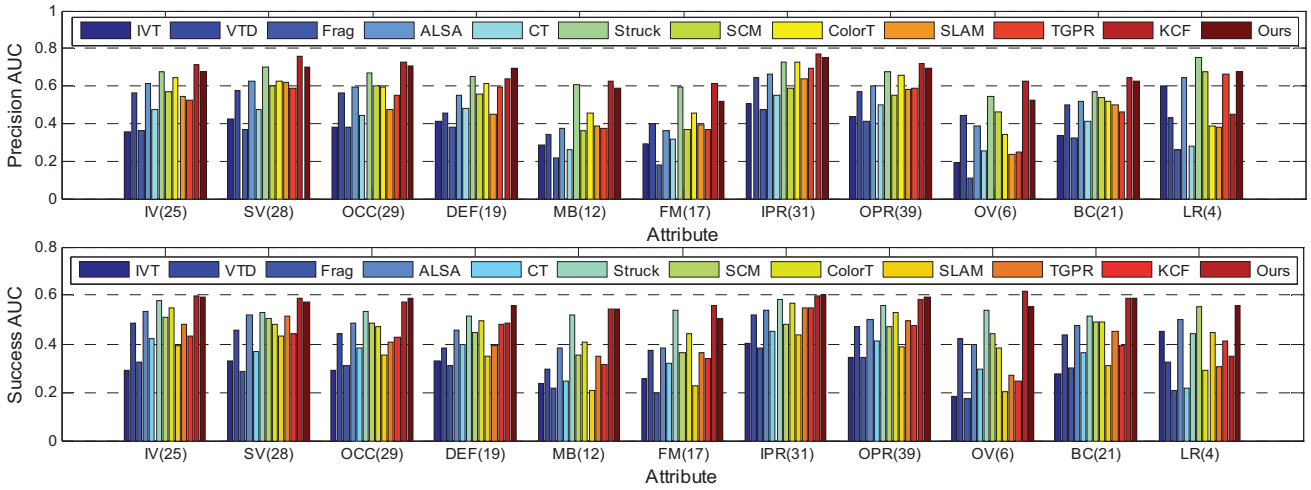


Figure 9. Precision AUC and success AUC of the 11 attributes [25] on the 51 benchmark videos. The 11 attributes are IV (illumination variation), SV (scale variation), OCC (occlusion), DEF (deformation), MB (motion blur), FM (fast motion), IPR (in-plane rotation), OPR (out-of-plane rotation), OV (out-of-view), BC (background clutters) and LR (low resolution) respectively. Under both the pixel center error criterion and the overlapping ratio criterion and on all the attributes, our method ranks among the top three. The number after each attribute is the corresponding video number.

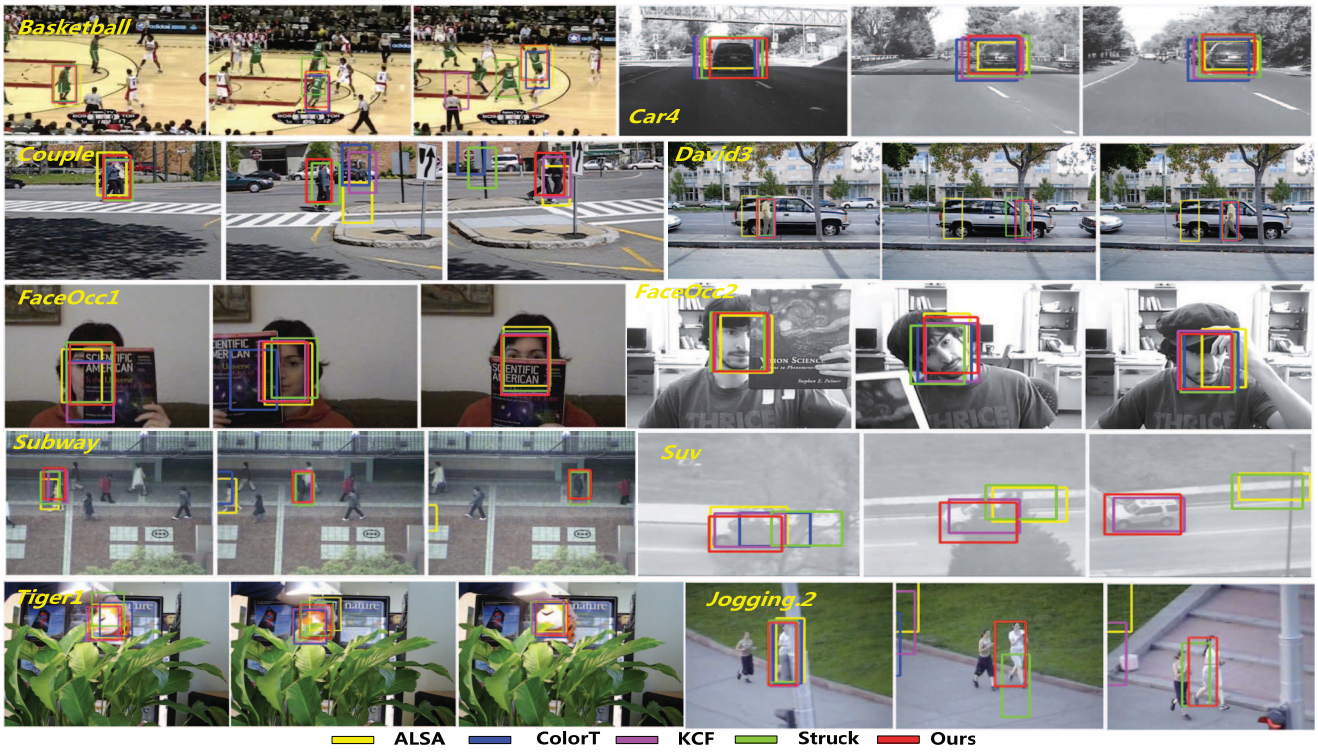


Figure 10. Sample frames of our method and four comparing methods on ten videos. The objects suffer from various challenges [25], such as drastic deformation and severe occlusion. Our method obtains promising results in comparison to other four methods.

[1] does not represent the nonlinear distribution of the samples and can not give accurate evaluation of the candidate states. SLAM [14] represents the nonlinear distributions of the samples. However, SLAM [14] can not retain the original Euclidean distance. In contrast, our method retains the original Euclidean distances between samples and also represents the nonlinear distribution of the samples. And then we obtain better performance than Frag [1] and SLAM [14].

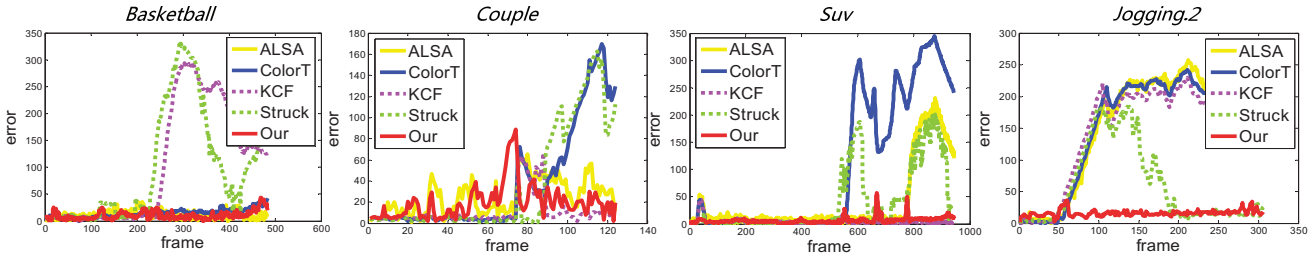


Figure 11. Pixel center error of our method and four comparing methods on four videos at each frame. Compared with other four methods, our method seldom loses tracks on the four videos.

Table 2. AUC values of two criteria of the comparing methods and our method. Red: best. Blue: second best.

	IVT	VTD	Frag	ALSA	CT	Struck	SCM	ColorT	SLAM	GTPR	KCF	Ours
Center	0.458	0.577	0.440	0.636	0.492	0.723	0.610	0.657	0.586	0.642	0.758	0.743
PASC	0.357	0.459	0.354	0.517	0.400	0.567	0.503	0.517	0.486	0.497	0.592	0.601

Qualitative Analysis: Our method obtains promising results when large challenges, such as occlusion, illumination variation, exist. Figure 10 shows some sample frames of our method tackling various challenges. Figure 11 shows the pixel center error of our method and four comparing methods on four videos at each frame. We divide the object appearance into four parts. When occlusion occurs, e.g. *FaceOcc1*, the non-occluded parts still have high evaluation values. And by combining the evaluations of the four parts together, our method still achieves robust performance.

When the object experiences drastic illumination variation, e.g. *Basketball* and *car4*, the appearance varies linearly to some extents. As our appearance model approximates the linear distribution, our method still tracks the object robustly.

Our model is able to represent the nonlinear distributions of the object samples when the object experiences drastic pose variation, e.g. *Basketball*. And thus our method is robust to object deformation. Besides, our model is also able to tackle other attributes when the object samples are distributed nonlinearly, e.g. inplane rotation. And by sampling the object size, our method also tackles scale variance. However, when severe background clutter occurs, e.g. *Ironman*, our method is disturbed and is not able to track the object robustly.

6. Conclusion and Future Work

In this paper, we have a local subspace collaborative tracking method for robust visual tracking. By combining the local linear subspace and local nonlinear subspace, more accurate evaluation of the object appearance is obtained. In the future, we will further investigate how to represent the nonlinear distributions of the samples with suitable geometries more effectively.

Appendix A

In this appendix, we give the computation process of obtaining (7). According to (5), we obtain

$$||\overline{S}_i - D_k||_2 = ||\overline{S}_j - D_k||_2, i, j = 1, ..., K. \tag{16}$$

According to (16), we obtain

$$\overline{S}_i^T \overline{S}_i - \overline{S}_j^T \overline{S}_j - 2(\overline{S}_i - \overline{S}_j)^T D_k = 0. \tag{17}$$

Based on (17) and (6), we form the objective function

$$g(D_k) = \sum_{i,j} w_i w_j ||\overline{S}_i^T \overline{S}_i - \overline{S}_j^T \overline{S}_j - 2(\overline{S}_i - \overline{S}_j)^T D_k||_2^2 + \lambda ||\overline{U}^T (\overline{S}_k - D_k)||_2^2. \tag{18}$$

By $\mathrm{d}g(D_k)/\mathrm{d}D_k = 0$, we obtain the result in (7). □

Acknowledgement

This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), and is supported by the Natural Science Foundation of China (Grant No. 61472421, 61225008, 61572271, 61527808, 61373074 and 61373090, 61472285, 6141101224, 61303178), the Project Supported by CAS Center for Excellence in Brain Science and Intelligence Technology, and the Project Supported by Guangdong Natural Science Foundation (Grant No. S2012020011081), the National Basic Research Program of China under Grant 2014CB349304, the Ministry of Education of China under Grant 20120002110033, and the Tsinghua University Initiative Scientific Research Program, Project of science and technology plans of Zhejiang Province (Grants No. 2015C31168).

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. *CVPR*, 2006. 2, 3, 5, 6, 7

[2] C. Alzate and J. A. K. Suykens. Multiway spectral clustering with out-of-sample extensions through weighted kernel pca. *PAMI*, 32(2):335–347, 2010. 1

[3] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. *CVPR*, 2009. 1

[4] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Tech. J.*, 2, 1998. 1, 2

[5] M. Danelljan, F. S. Khan, M. Felsberg, and J. Weijer. Adaptive color attributes for real-time visual tracking. *CVPR*, 2014. 5, 6

[6] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. *ECCV*, 2014. 2, 5

[7] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. *ICCV*, 2011. 5, 6

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 24(5):1–14, 2015. 2, 5, 6

[9] M. Isard and A. Blake. Condensation c conditional density propagation for visual tracking. *IJCV*, 1(29):5–28, 1998. 2

[10] X. Jia, H. Lu, and M. H. Yang. Visual tracking via adaptive structural local sparse appearance model. *CVPR*, 2012. 1, 2, 5, 6

[11] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 34(7):1409–1422, 2012. 2

[12] J. Kwon and K. M. Lee. Visual tracking decomposition. *CVPR*, 2010. 5

[13] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. *BMVC*, 2014. 2

[14] X. Li, W. Hu, Z. Zhang, and X. Zhang. Robust visual tracking based on an effective appearance model. *ECCV*, 2008. 2, 3, 5, 7

[15] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Visual tracking via incremental log-euclidean riemannian subspace learning. *CVPR*, 2008. 2, 3

[16] X. Mei and H. Ling. Robust visual tracking using l1 minimization. *ICCV*, 2009. 2

[17] D. A. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008. 1, 2, 3, 5

[18] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323– 2326, 2000. 2, 3

[19] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. *CVPR*, 2012. 1, 2

[20] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 2, 3

[21] P. Tiwari, J. Kuihanewicz, M. Rosen, and A. Madabhushi. Semi supervised multi kernel (sesmik) graph embedding: Identifying aggressive prostate cancer via magnetic resonance imaging and spectroscopy. *MICCAI*, 2010. 1, 3

[22] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cogn. Neurosci.*, 3(1):71–86, 1991. 1

[23] H. Z. Wang, D. Suter, K. Schindler, and C. H. Shen. Adaptive object tracking based on an effective appearance filter. *PAMI*, 29(9):1661–1667, 2007. 2

[24] N. Wang, J. Wang, and D. Yeung. Online robust non-negative dictionary learning for visual tracking. *ICCV*, 2013. 2

[25] Y. Wu, J. Lim, and M. Yang. Online object tracking: A benchmark. *CVPR*, 2013. 2, 5, 7

[26] Y. Wu, B. Ma, M. Yang, Y. Jia, and J. Zhang. Metric learning based structural appearance model for robust visual tracking. *TCSVT*, 24(5):865–877, 2014. 2

[27] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *PAMI*, 29(1):40–51, 2007. 3

[28] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. Hengel. Part-based visual tracking with online latent structural learning. *CVPR*, 2013. 2

[29] H. Zhang, Z. Zha, S. Yan, M. Wang, and T. Chua. Robust non-negative graph embedding: Towards noisy data, unreliable graphs, and noisy labels. *CVPR*, 2012. 3

[30] K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. *ECCV*, 2012. 5

[31] X. Zhang, W. Hu, S. Chen, and S. Maybank. Graph-embedding-based learning for robust object tracking. *TIE*, 61(2):1072–1084, 2014. 1

[32] Z. Y. Zhang and H. Y. Zha. Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM J. Sci. Comput.*, 26(1):313–338, 2004. 3

[33] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. *CVPR*, 2012. 5