Robust Visual Tracking Using Local Sparse Appearance Model and $K$-Selection

Baiyang Liu, Member, IEEE, Junzhou Huang, Member, IEEE, Casimir Kulikowski, Fellow, IEEE, and Lin Yang†, Member, IEEE

Abstract—Online learned tracking is widely used for its adaptive ability to handle appearance changes. However, it introduces potential drifting problems due to the accumulation of errors during the self-updating, especially for occluded scenarios. The recent literature demonstrates that appropriate combinations of trackers can help balancing the stability and flexibility requirements. We have developed a robust tracking algorithm using a local sparse appearance model (SPT) and $K$-Selection. A static sparse dictionary and a dynamically updated online dictionary basis distribution are used to model the target appearance. A novel sparse representation-based voting map and a sparse constraint regularized mean-shift are proposed to track the object robustly. Besides these contributions, we also introduce a new selection based dictionary learning algorithm with a locally constrained sparse representation, called $K$-Selection. Based on a set of comprehensive experiments, our algorithm has demonstrated better performance than alternatives reported in the recent literature.

Index Terms—Sparse Representation, Tracking, $K$-Selection, Appearance Model, Dictionary Learning

1 INTRODUCTION

Visual tracking estimates the spatial state of a moving target through observed sequences. This topic is interesting and significant in many industrial applications including security surveillance, traffic monitoring, vehicle navigation, video index, robotics, etc. Even though there is substantial literature proposing to tackle this problem for decades, accurate tracking of general objects in a dynamic environment still presents difficulties due to the following challenges [36], [37]:
- Dynamic appearance changes due to illumination, rotation, and scaling;
- 3D pose variations and information loss due to the projection from 3D to 2D;
- Partial and full object occlusions;
- Complex background clutter;
- Similar objects from the same class which lead to landmark ambiguities.

Generative and discriminative methods are two major categories used in current tracking techniques. The generative models formulate the tracking problem as a search for the regions with the highest likelihood [5], [8], [23], [26], [27], [33], [39], [22], [16]. To address the target appearance changes in a dynamic environment, it has also been proposed to keep updating the target appearance model incrementally to adapt it to appearance changes.

An incrementally updated linear subspace was proposed to model the target appearance in [27], while the parameters of the target appearance model were online adapted online using the EM algorithm in [16]. As single appearance model is argued to be insufficient to represent the target in a dynamic environment, multiple appearance models have been proposed to be incrementally learned so as to compensate for the non-linear appearance changes [38]. Discriminative methods formulate tracking as a classification problem [3], [4], [6], [14]. The trained classifier is used to discriminate the target from its background, and it is also updated online. Grabner et. al. [11] proposed to update the selected features incrementally using current tracking results, which may lead to potential drifting because of accumulated errors. The errors are aggravated when there exist occlusions.

In order to handle the drifting problem, semi-online boosting [12] was later proposed to incrementally update the classifier using both unlabeled and labeled data. The Multiple Instance Learning boosting method (MIL) [4] puts all samples into bags and labels them. The positive bag is required to contain at least one real positive, while the negative bags have only negative samples. The drifting problem is handled in this method since the true target included in the positive bag is learned implicitly. Instead of tackling the problem with new learning methods, recently, it has been shown that an appropriate combination of complementary tracking algorithms can help alleviate drifting problems [17], [34], [35], [29], [28]. Simple online learner has two extreme counterparts in terms of adaptivity and stability. More stable tracker can moderate and guide the updating of active learner. However, well balancing of the stability and flexibility dilemma is difficult in general without an efficient way to combine trackers.
The target appearance (a) is modeled with a dictionary (b) and a sparse coding histogram (c). The confidence map (e) of the image (d) is the inverse of the reconstruction error from the learned target dictionary. The target center is found by voting and sparse constraint with regularized mean-shift on the probability map (f).

The related work on tracking using sparse representation is reviewed in Section 2. The local appearance modeling is introduced in Section 3. In Section 4, we describe a novel selection based dictionary learning method for sparse representation with better discriminative capability than K-SVD [2]. The tracking procedure with voting and mean-shift is presented in Section 5. Section 6 represents the comparative experiment conducted on public tracking test benchmarks with the most recent state-of-the-art methods. Finally, Section 8 concludes the paper.

2 Related Work

Recently, it has been noted that finding an effective way to handle occlusion is important for both accurate tracking and robust online learning, because ambiguities between occlusion and appearance changes may mislead an online learner. In the recent literature, the learned sparse representation has been utilized in many areas [13, 7, 21, 32] and successfully applied for tracking [24, 19, 18]. The tracking problem is formulated as finding a sparse approximation in the template subspace \( \Phi \).

With the sparseness assumption, the given signal \( x \) can be represented as the linear combination of a few basis vectors from the collected library \( \Phi \) with component \( f \) representing the noise:

\[
x = \Phi \alpha + f.
\] (1)

The representation coefficient \( \alpha \) can be computed by optimizing the \( l_1 \) regularized least square problem, which typically provides a sparse solution [32]:

\[
\alpha^* = \text{argmin}_{\alpha} \| x - \Phi \alpha - f \|^2 + \lambda \| \alpha' \|_1
\] (2)

where \( \alpha' = (\alpha^T, f^T)^T \) and the parameter \( \lambda \) controls the sparsity of both coefficient vector and noise.

The results are found to be efficient and adaptive to appearance changes, especially occlusion. In [19], Dynamic group sparsity was utilized to favor a group selection of features. Two stage sparse optimization was proposed for efficiency which provides more discriminative power than [24]. All these methods model the target as a single entity, and therefore cannot handle partial occlusion very well. Fragment-based tracking as in [1] coupled with a voting map can accurately track the partially occluded target. However, this method tracks each target patch with a static template, which limits its expressive power. It may fail in a dynamic environment which exhibits appearance changes or pose variations.

In this paper, we have proposed a robust tracking algorithm with a local sparse appearance model (SPT). The algorithm’s key components are a static sparse dictionary which is used to limit drifting and keep the flexibility in its linearly spanned subspace; a dynamic dictionary basis distribution is represented by a sparse coding histogram and updated online; a sparse representation-based voting map and reconstruction error regularized mean-shift are used to locate the center of the object as a final step. Besides all these contributions, a novel sparse dictionary learning method, called \( K \)-selection, is introduced to learn the target’s sparse representation library. Figure 1 illustrates an overview of the proposed algorithm. To our knowledge, this is the first contribution to tracking with online learned fragment-based sparse representation using a static basis but a dynamic structure; and \( K \)-selection is also the first selection based dictionary learning method for sparse representation. The contributions of this paper are:

- A natural combination of static sparse dictionary and dynamic online updated basis distribution considering both adaptivity and stability.
- A novel selection based sparse dictionary learning method by directly selecting the most representative basis.
- A sparse representation-based voting map and sparse constraint regularized mean-shift for object tracking.

3 Local Sparse Appearance Model

In our algorithm, a local sparse representation is used to model the appearance of target patches, and the sparse coding histogram represents the basis distribution of the target.

3.1 Local Sparse Representation

Given an image \( I \) of the current frame, we can sample a set of small image patches \( X = \{ x_i | i = 1 : N \} \) centered at each pixel inside the target region by sliding a window of size \( m \times n \), where \( x_i \) is the \( i \)th column vectorized image patch. If \( p = m \times n \) is its dimension, and \( \Phi \in \mathbb{R}^{p \times k} \) is the basis dictionary learned from the target patches, the target patches can be reconstructed by solving Eqn. (2). Many methods have been proposed to optimize this problem, such as orthogonal matching pursuit (OMP) [30], subspace pursuit [9], dynamic group sparse [15], etc. In tracking applications, since similarity is more essential than sparsity, locality-constrained linear coding (LLC) [31] is...
utilized in our algorithm for solve the sparse representation problem. In LLC, the objective function is formulated as
\[
\min_\alpha ||x - \sum_{i=1}^{K} \Phi_i \alpha_i||^2 + \lambda ||d \odot \alpha||^2,
\]
\[
s.t. 1^T \alpha = 1
\]
where $\odot$ is element-wise multiplication and $d$ is exponential to the Euclidean distance vector between $x$ and all basis vectors in $\Phi$. The constraint $1^T c = 1$ here is used to ensure the shift-invariance. The solution $\alpha$ is not $l_0$ norm sparse, but has only a few significant components.

Since the second term in Eqn. (3) is exponential in the distance from the candidate patch to the basis and thus leads to a very large penalty for the components which are far away from the candidate. LLC actually selects a set of local basis vectors for $x$ to form a local coordinate system. Thus, a faster approximate LLC can be derived by solving a smaller linear system $B$ containing the $k$ nearest neighbors of $x$ [31].
\[
\min_\hat{\alpha} ||x - B \hat{\alpha}||^2,
\]
\[
s.t. 1^T \hat{\alpha} = 1.
\]

Since $B$ is a subset of $\Phi$ and $\hat{\alpha}$ in Eqn. (4) is a dense solution, $\hat{\alpha}$ can be considered as the non-zero coefficients in the original $\alpha^*$ of LLC corresponding to the components in $B$ with the coefficients of all other components as 0. In terms of tracking, LLC is used to compute the representation with the local template patches (basis) in $\Phi$ that are similar to the candidate sample $x$.

### 3.2 Sparse Coding Histogram

With the sparse representation model presented above, the local patches from target regions should have smaller reconstruction errors than the background clutter. But as illustrated in Figure 1(e), some regions of the book, which belong to background, have relatively large probability since their appearance falls within the subspace spanned by the target dictionary basis. A few contaminated bases selected from the background due to an inaccurate initial rectangle may affect the performance of tracking, especially in the static camera scenario where target appearance changes, but background remains static.

Thus, more structural information is necessary for accurately identifying the target. In this section, we propose a sparse coding histogram to represent the appearance distribution of the target model and derive sparse constraint regularized mean-shift tracking methods which will be further discussed in Section 5.

For each image patches $x_i$ in the template, let $\alpha_i$ as the optimal sparse coefficients in Eqn. (3). The coding histogram is defined as the coefficients sum of the coefficients of basis with non-zero coefficient values
\[
H(b_j) = \sum_{i=1}^{N} |\alpha_{ij}|
\]
where $\alpha_{ij}$ is the $j$-th coefficient of the $i$-th image patch. Similar to a color histogram, the sparse coding histogram indicates how the appearance dictionary basis is distributed on the target model.

**Target Model**: Let the target center be taken as the origin of the model frame of reference. Define $x_i$, $i = 1 : N$ as the vectorized image patches centered at pixel position $c_i$, an isotropic kernel $k(c_i)$ is applied to assign smaller weights to patches far away from the center. The value of the $j$-th bin $q_j$ in the target model can be computed as a weighted sum:
\[
q_j = C \sum_{i=1}^{N} k(||c_i||^2) |\alpha_{ij}|
\]
where $C$ is a constant to ensure $\sum_{i=1}^{N} q_j = 1$.

**Target Candidate**: Define $x'_i$, $i = 1 : N'$ as the vectorized image patches centered at pixel position $c_i$ inside the window centered at $y$. The value of the $j$-th bin, $\hat{p}_j(y)$, in the candidate model can be computed as:
\[
\hat{p}_j(y) = C \sum_{i=1}^{N'} k(||y - c_i||^2) |\alpha^*_j|
\]
where $\alpha^*$ is the solution of Eqn. (3) and $h$ is the scale factor.

The sparse coding histogram is dynamic when a target experiences variations and is updated online. Let $y$ be the new target center found in the current frame and $\hat{p}_j(y)$ as its coding histogram from Eqn. (7), the new appearance basis histogram can be updated with learning rate $\gamma$:
\[
q'_j = q_j(1 - \gamma) + \hat{p}_j(y) \gamma
\]

### 4 Dictionary Learning by $K$-Selection

We assumed that the dictionary $\Phi$ is given in all the above. Many methods have been proposed to learn a dictionary to minimize the overall reconstruction error in sparse representations [10], [20], [2]. The target templates are stored and updated to form the dynamic dictionary in [24], [19].

Here, we introduce a new method to learn the dictionary as basis selection by gradient descent. Given a dataset $X = \{x_i | i = 1 \ldots N\}$, the problem can be formulated as selecting $K$ data vectors as a basis from the dataset which minimizes the objective function:
\[
f(\Phi) = \sum_{i=1}^{N} ||x_i - \sum_{k=1}^{K} \Phi_k \alpha_{ik}||^2 + \lambda ||d_i \odot \alpha_i||^2,
\]
\[
s.t. 1^T \alpha_i = 1, \forall i
\]
where $\Phi_k = x_{b_k}$ and $b_k$ is the index of data vector selected as the $k$-th basis vector. The $d_i$ is exponential to the Euclidean distance from $x_i$ to the dictionary as in Eqn (3) and $\alpha_i$ the representation coefficients. Exhaustive search is needed to find the optimal solution, so we propose an efficient method which can quickly converge to a suboptimal solution, called $K$-Selection. To our knowledge, it is the first paper to formulate the selection-based dictionary learning problem for sparse representation.
4.1 Basis Initialization

The initial set of basis vectors is chosen by the following criterion. For any data point \( x_i \in X \), the sparse representation with all other points as a dictionary can be solved by

\[
\min_{\omega_j} \| x_i - \sum_{j=1,j \neq i}^N x_j \omega_{ij} \|^2 + \lambda \| d_i \odot \omega_i \|^2, \\
\text{s.t.} \quad 1^T \omega_i = 1,
\]

where \( \omega_{ij} \) indicates the importance of the \( j \)-th data point for sparsely representing the \( i \)-th data point. Thus, for the \( j \)-th point, the importance weight of \( w_j \) to be selected as a basis vector is

\[
w_j = \sum_{i=1}^N |\omega_{ij}| e^{-\frac{\epsilon_i^2}{\sigma^2}}.
\]

The reconstruction error \( \epsilon_i \) indicates the reliability of this representation. In other words, the importance of the \( j \)-th data point is its weighted contribution in representing the entire dataset. The first \( K \) data vectors with the largest \( w \) are selected as the initial basis.

4.2 Gradient Descent

After initialization, a new data vector will be selected to replace the \( t \)-th basis vector to minimize the cost function iteratively. Let \( \alpha_i \) be the LLC result for \( x_i \) with the current \( \Phi \) and, setting low-value components to zero, the dictionary is updated to fit the dataset without the locality constraint. The gradient with respect to the \( t \)-th basis can be approximated as

\[
\nabla f_t = \frac{\partial f}{\partial \Phi_t} = -2 \sum_{i=1}^N (x_i - \sum_{k=1}^K \Phi_k \alpha_{ik}) \alpha_{it}.
\]

Instead of directly updating the basis in the direction of the negative gradient \( r_t = -\nabla f_t \), we perform the update by selecting the data point \( x_t \) which has the largest correlation between the displacement and the \( t \)-th basis

\[
COR(x_t, x_b, r_t) = \frac{(x_t - x_b)^T r_t}{|| (x_t - x_b) ||_2 |r_t|_2}.
\]

The data point \( x_t \) with the maximal value of \( COR \) is selected as a potential candidate to replace the \( t \)-th basis. Let \( f_{min} \) be the current residual and \( f_{rep} \) as the residual after replacing the \( t \)-th basis with \( x_t \), then the replacement will be done only if \( f_{min} > f_{rep} \).

Compared to other dictionary learning methods, such as \( K \)-SVD, the dictionary learned with \( K \)-Selection has a constrained capability to represent the dataset. However, the target library learned with \( K \)-SVD is so general that some of the background image patches can also be well represented. This is not desirable in visual tracking, which requires strong discriminative ability. In order to provide higher discriminative power, we limit the space spanned by the learned target library strictly to the target model itself, by directly selecting the \( K \) data vectors from the dataset. This discussion will be revisited in Section 6.1.

5 TRACKING

5.1 Sparse Constraint Regularized Mean-shift

In this section we present an iterative tracking algorithm to locate a target with a local appearance model. Let \( y \) be the target center candidate, while \( X = \{ x_i, i = 1...N \} \) represents \( N \) patches in the window \( W \) centered at \( y \). Tracking is aimed at locating the target with maximum generative likelihood, and match the target model and candidate models.

The probability of \( y \) being the target center can be estimated by the products of the probability of each candidate patch within \( W \) as potential target patches,

\[
P(y|\Phi) = C \prod_{i=1}^N e^{\frac{-k(|| \frac{y - c_i}{h} ||_2^2)}{\sigma^2}},
\]

where \( \epsilon_i \) is the sparse reconstruction error for the \( i \)-th patch. The log probability of the target candidate is then:

\[
L(y|\Phi) = \sum_{i=1}^N -k(|| \frac{y - c_i}{h} ||_2^2) \epsilon_i^2.
\]

The Bhattacharyya metric is used to measure the distance between the sparse coding histograms of the target and candidate models

\[
d(y) = \sqrt{1 - \rho(\hat{p}(y), q)},
\]

\[
\rho(\hat{p}(y), q) = \sum_{j=1}^K \sqrt{\hat{p}_j(y)q_j}.
\]

Considering that the similarity of the target to the learned dictionary and the similarity to the dictionary basis distribution are independently conditioned on the same candidate, the overall objective function can be formulated as:

\[
\hat{\rho}(y, \Phi) = \sum_{j=1}^K \sqrt{\hat{p}_j(y)q_j} L(y|\Phi).
\]

The first component in \( \hat{\rho}(y, \Phi) \) measures the match between the distribution of the target model and candidate model, while the second term measures the probability of the candidate being generated from its target library \( \Phi \).

Assume that we have an initial guess of the center as \( y_0 \). With the assumption of small perturbation of \( y \) from \( y_0 \), Eqn. (18) can be rewritten based on Taylor expansion:

\[
\hat{\rho}(y, \Phi) \approx -\frac{1}{2} \sum_{j=1}^K \sqrt{\hat{p}_j(y_0)q_j} L(y_0|\Phi) + \sum_{j=1}^K \sqrt{\hat{p}_j(y_0)q_j} L(y|\Phi)
\]

\[
+ \frac{1}{2} \sum_{j=1}^K \hat{p}_j(y) \sqrt{\frac{q_j}{\hat{p}_j(y_0)}} L(y_0|\Phi)
\]

\[
= C_1 + \frac{1}{2} \sum_{i=1}^N w_i k(\frac{y - c_i}{h} ||_2^2),
\]

where \( k(x) \) is the kernel function that measures the similarity between \( x \) and the center.
The target center can be found iteratively using Mean-shift term in Eqn. (19) has to be maximized to minimize the probability of the target location is represented by different red color level, with lighter colors indicating lower probability values. The image patches labeled with red rectangle use the selected basis (with non-zero coefficient) for the sparse representation. Each of them contributes to the probability map of the basis weighted by the coefficients in Eqn. (22). Let the position of the dictionary patch as the coordinate origin (green dot in the right figure), the probability of the target location is represented by different red color level, with lighter colors indicating lower probability values. We can see that the contribution from patches in region a is smaller than those from region b, since their similarity to the basis is smaller than the ones in region b demonstrated by their coefficients. In this way, the probability map modeled in the dictionary can be utilized to compute the final voting map to find the target center through spare representation.

**Tracking stage:** Denote $x_i$ as the $i$-th image patch centered at $c_i$ and $\alpha_j^i$ as its coefficients calculated by LLC. The overall target center voting map $V$ can be computed as:

$$V(c) = \sum_{i=1}^{N} \sum_{j=1}^{K} P_i(c - c_i, j)(1 - \delta(\alpha_j^i))e^{-\frac{r^2}{\sigma^2}},$$  

(23)

where $\delta(x)$ is Dirac delta function. Only the probability map of those dictionary patches with non-zero coefficients will contribute to the final voting map. The $e^{-\frac{r^2}{\sigma^2}}$ in Eqn. (23) weights the voting by its sparse reconstruction accuracy. Patches with larger errors contribute less to the overall voting map. The details of the voting algorithm in the tracking stage are given in Algorithm 1. Using the voting map, the final tracking result can be found iteratively with

$$\hat{y}_{t+1} = \frac{\sum_{j=1}^{N} c_i w_i V(c_i) g(||\hat{y}_t - c_i||^2)}{\sum_{j=1}^{N} w_i V(c_i) g(||\hat{y}_t - c_i||^2)}$$

(24)

The detailed procedures of our tracking framework are listed in Algorithm 2.

**6 Experiment**

In this section, we evaluate our sparse tracking algorithm (SPT) on eight challenging sequences and compare its performance with five other recent state-of-the-art trackers. Besides, a set of experiments on celebrity face data and dynamic sequences with large amount of motion were also tested to demonstrate the performance of SPT. For the
3D Pose

Occlusion

Scaling

Illumination

and stable results in these standard testing benchmarks for overall, our proposed method provides the most accurate parameter selections is presented in Section 6.2, followed by the comparative tracking results shown in Section 6.3. The challenges of these sequences are summarized in Table 3, including pose variation, illumination changes, occlusions and scaling.

Section 6.1 demonstrates the performance of the proposed K-Selection dictionary learning method in terms of generative and discriminative capability. The discussion of parameter selections is presented in Section 6.2, followed by the comparative tracking results shown in Section 6.3. Overall, our proposed method provides the most accurate and stable results in these standard testing benchmarks for tracking.

6.1 Reconstruction vs. Discriminative Power

As denoted in Section 4, it is worthwhile to evaluate not only the reconstruction error, but also the discriminative power for a sparse dictionary learning method. We claim that under certain conditions, discriminative power weights more than overall reconstruction error. In this set of experiments, more than a hundred thousand image patches were extracted from the target region, and the same number of background patches were randomly generated from the regions outside the target. The dictionaries were trained using the target patches extracted from the first frame only.

If $X^+$ and $X^-$ are the set of target patches and background patches, the reconstruction error is measured as

$$E(X) = \frac{1}{N} \sum_{i=1}^{N} ||x_i - \Phi \alpha_i^*||$$

where $\alpha_i^*$ is the sparse solution of the $i$-th patch using Eqn. (3). The difference between $E(X^+)$ and $E(X^-)$ is used to measure the discriminative power of the learned dictionary. A larger difference $|E(X^+) - E(X^-)|$ indicates stronger discriminative power.

The popular K-SVD method [2] with orthogonal matching pursuit (OMP) [30] is used for comparison. As shown in Figure 6.1(a), it is not surprising that the dictionary learned with K-SVD has a smaller overall reconstruction error than our K-Selection method as it explicitly minimizes the $l_2$ reconstruction errors. However, the dictionary learned by K-SVD with OMP can also represent the background patches, which leads to relatively weaker discriminative power compared with K-Selection, as shown in Figure 6.1(b). Therefore, K-Selection exhibits larger reconstruction errors but stronger discriminative power, which makes it more suitable for tracking and some other applications. Also, discriminative power does not always increase by adding more basis patches, because the added basis will contribute to the reconstruction of both background and target. We found that a very small basis set is often sufficient to discriminate target from background.

6.2 Parameter Analysis

The proposed algorithm has two important parameters: patch size $s$ and percentage $\beta$ of the selected basis over the

TABLE 3

The challenges of the experimental sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>3D Pose</th>
<th>Illumination</th>
<th>Occlusion</th>
<th>Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>David</td>
<td>√‡</td>
<td>√</td>
<td>×</td>
<td>√</td>
</tr>
<tr>
<td>girl</td>
<td>√‡</td>
<td>√†</td>
<td>√‡</td>
<td>√</td>
</tr>
<tr>
<td>car</td>
<td>×</td>
<td>√</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>faceocc2</td>
<td>×</td>
<td>√</td>
<td>√‡</td>
<td>×</td>
</tr>
<tr>
<td>lemmin</td>
<td>√</td>
<td>×</td>
<td>√‡</td>
<td>√</td>
</tr>
<tr>
<td>board</td>
<td>√‡</td>
<td>×</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>liquor</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

† Partial variation or occlusion.


TABLE 1

Algorithm 1: Compute sparse representation based voting

**Define:** $x_i$ as the $i$th image patches centered at position $c_i$

1. Initialize $V = 0$.
2. for $i = 1 : N$
3. $\alpha_i^* \leftarrow$ solution of LLC Eqn. (3)
4. $c_i = ||x_i - \Phi \alpha_i^*||_2$
5. for $j = 1 : K$
6. for all locations $c$
7. $V(c) = V(c) + P(c - c_i, j)(1 - \delta(c - c_i))e^{-\frac{x^2}{2}}$
8. end
9. end
10. end

Algorithm 2: Tracking Framework Summary

**Input:** Target template

**Training Stage**

1. Extracting target samples from target template as training data $X$
2. Learning the target patch dictionary $\Phi$ using K-Selection in Sec.4
3. Solving the sparse representation for each training data by optimizing Eqn. (3)
4. Compute the target appearance model discussed in Sec. 3.2
5. Encoding the target spatial information for each dictionary component using Eqn. (22)

**Tracking Stage**

1. Set the tracking result in the last frame as the initial guess of target center in current frame
2. For different scale changes in the range
3. Do image warping for given scale and rotation to get $I'$
4. Extracting the candidate samples from $I'$
5. Computing the coefficients for each sample using Eqn. (3)
6. Compute the voting map of target center with Eqn. (23)
7. Estimate the new target center with mean-shift iteratively
8. End
9. The tracking result with highest score and corresponding scale and rotation will be considered as the final tracking result
10. Projecting the tracking result back to original image frame by doing affine transformation with tracked scale and rotation
11. Update the target model and voting map if online update enabled

TABLE 2

Summary of overall SPT tracking framework

whole training set. Four sequences (David, girl, faceocc2, board), exhibiting illumination changes, pose variations and occlusions were tested with 
$s = (3 \times 5, 7 \times 7, 9 \times 9, 11 \times 11)$ and 
$\beta = (5\%, 10\%, 15\%, 20\%, 25\%, 30\%)$.

As we can observed in Figure 4(a), patch sizes 5 and 7 provide the best results. A dictionary learned with smaller image patches have more representation ability but less discriminative power. Similar trends can be observed for the percentage of selected basis vectors over the entire training set, shown in Figure 4(b). A larger dictionary will deteriorate tracking performance due to the loss of discrimination.

6.3 Comparative Tracking Results

In this section, the performance of the proposed tracking algorithm (SPT) was compared with other recent methods. 

**Benchmark Sequences:** SPT was first evaluated on four benchmark sequences compared with: multiple instance learning (MIL) [4], online simple tracking (PROST) [28], two stage sparse tracker (TST) [19] and incremental visual tracking (IVT) [27]. For a fair comparison, the mean ratio of a target center’s offset over the diagonal length of the target is used to measure the performance. The quantitative results are shown in Table 4. The best tracking results were emphasized as bold. Our method produces the smallest tracking offset (measured by the Euclidean distance from the center of the target to the ground-truth) are shown in Figure 5(a). It is clear to see that SPT generate very stable result with smallest tracking error in this sequence.

Figure 7 presents the tracking results of the alternative methods for a Girl sequence. The challenges of this sequence include 360 degree 3D pose variations and occlusions by another face which is similar to the target. Even though our method (using static appearance dictionary) is not designed to handle large pose and appearance variations, it generates good result because small parts from the target can still produce higher voting score to differentiate the target center from the background. The tracking offsets of SPT become slightly larger for frames with quick pose variations (e.g. #314), because the encoded target configuration is not exactly matched. From Figure 5(b), we can see that MIL can generate stable results for this sequence through online updating, but it still cannot handle large and fast pose variations. IVT has more accurate results for those frames (e.g. #314) with rotation. However, the single, online updated subspace of IVT drifted to track another face as shown in frame #436. SPT generates very good result in heavy occlusion scenario such as frame #436.

The Car sequence was captured in an open road environment. The tracking results of the #32, #180. #200, #230, #240 and #352 are presented in Figure 8. The MIL starts to show some target drifting (on the #200 frame) and finally loses the target (the #240 frame). IVT can track this sequence quite well. The TST can accurately track the car too, but the bounding box is not as accurate as our method and IVT. The target was successfully tracked using the proposed SPT during the entire sequence. The detailed quantitative performance for Car sequence is shown in Figure 5(c).

The Face Occlusion sequence was used to test the robustness of the proposed algorithm in handling occlusion. The #80, #150, #208, #540, #722 and #741 frames are presented in Figure 9. The MIL algorithm can roughly capture the position of the object, but did show some drifting problems when there is heavy occlusion, shown in the #741 frame. IVT start to fail after the #540 frame. SPT and TST provide reasonable results. Similar to the car
sequence, SPT still provides more accurate bounding box than TST. The detailed quantitative performance for this sequence is shown in Figure 5(d).

**PROST Sequences:** To further evaluate the proposed SPT for object tracking under occlusion, appearance blur, and pose variation, the latest four sequences provided in [28] are selected to compare with PROST, MIL, and FragTracker [1]. The proposed SPT produced the best performance for all sequences, as shown in Table 5, while PROST has the second best performance. Pixel-wise tracking results and the results of selected frames show that other methods have difficulties in accurately locating the target under heavy occlusion, as we show in the #336 frame in lemming sequence (Figure 10), the #300 in the box sequence (Figure 11), and in the #731 frame in the liquor sequence (Figure 13(d)). MIL and PROST cannot track the target accurately when large pose variation occurs, as we show in the #994 frame in the lemming sequence (Figure 10), the #600 frame in the box sequence (Figure 11), the #497 frame in the board sequence (Figure 13(b)), while our SPT method can track the target even under 90 degree off-plane rotation, as we shown in the #497 frame in the board sequence (Figure 13(a)) and the #731 frame in the liquor sequence (Figure 13(c)).

**Celebrity Face Data:** In Figure 14, tracking results of selected frames from a celebrity face dataset in [25] are presented here. In this sequence there are multiple expression and 3D pose variations, as well as scale changes. As we can see from the results, SPT can generate accurate

### Table 4

<table>
<thead>
<tr>
<th></th>
<th>david</th>
<th>girl</th>
<th>car</th>
<th>faceocc2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROST[28]</td>
<td>0.124</td>
<td>0.115</td>
<td>NA</td>
<td>0.116</td>
</tr>
<tr>
<td>TST[19]</td>
<td>0.052</td>
<td>0.131</td>
<td>0.065</td>
<td>0.139</td>
</tr>
<tr>
<td>MIL[4]</td>
<td>0.127</td>
<td>0.161</td>
<td>0.700</td>
<td>0.095</td>
</tr>
<tr>
<td>IVT[27]</td>
<td>0.059</td>
<td>0.147</td>
<td>0.020</td>
<td>0.081</td>
</tr>
<tr>
<td>SPT</td>
<td><strong>0.026</strong></td>
<td><strong>0.066</strong></td>
<td><strong>0.031</strong></td>
<td><strong>0.065</strong></td>
</tr>
</tbody>
</table>
Fig. 7. The tracking results of selected frames on Girl sequence using the proposed method (SPT), Multiple Instance Learning (MIL), Two Stage Tracker (TST) and Incremental Visual Tracker (IVT).

Fig. 8. The tracking results of selected frames on Car sequence using the proposed method (SPT), Multiple Instance Learning (MIL), Two Stage Tracker (TST) and Incremental Visual Tracker (IVT).

### TABLE 5
Average comparative results on the PROST datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>lemming</th>
<th>box</th>
<th>board</th>
<th>liquor</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROST[28]</td>
<td>0.189</td>
<td>0.091</td>
<td>0.157</td>
<td>0.101</td>
</tr>
<tr>
<td>FragTrack[1]</td>
<td>0.625</td>
<td>0.406</td>
<td>0.363</td>
<td>0.145</td>
</tr>
<tr>
<td>MIL[4]</td>
<td>0.112</td>
<td>0.740</td>
<td>0.206</td>
<td>0.619</td>
</tr>
<tr>
<td>SPT</td>
<td>0.101</td>
<td>0.073</td>
<td>0.059</td>
<td>0.016</td>
</tr>
</tbody>
</table>

**Dynamic Motion Sequences**: To demonstrate that our method works in a dynamic scenario, the proposed algorithm is tested on three additional sequences (animal, football, and basketball), presented in Figure 15. The targets in all three videos have large motions. The top row in Figure 15 represents the tracking result of an animal sequence. The target was running on the water with other animals exhibiting similar appearance. SPT generates accurate results even with heavy motion blur and water occlusion. The middle row in Figure 15 presents the tracking results of a football sequence with large motions. The bottom row in Figure 15 shows the tracking results of a basketball sequence with fast movements.
game sequence with a large amount of similar background clutters. It can be observed that the proposed SPT can follow the football player along the whole sequence very well. The basketball sequences shown in the bottom row demonstrated that our patch-based voting methods can handle non-rigid target with a reasonable dynamic spatial configuration. However, a large, non-rigid transformation might fail the proposed tracking method, since the learned target center configuration relative to each part will not be valid anymore. An adaptive tracking using online update voting map will be discussed in Section 7.

7 DISCUSSION

In this section, we will discuss potential shortcomings of the method, possible ways of overcoming them, and future work.

Potential Failures In this paper, we proposed a general tracking framework to integrate both the appearance model and spatial configuration of the target. However, only the relative target center information is encoded into the dictionary and then retrieved for locating the target along the sequence. There are some potential failure cases where the method may fail due to this limitation.

- Large movement between consecutive frames might lead to potential failure of SPT since the target center is found by iteratively moving from the initial guess to the final solution. However, relatively small perturbations between consecutive frames will not present a problem.
- Unseen appearance changes will reduce the validity of the learned appearance model and lead to possible errors. To balance the stability and flexibility, the dictionary in SPT is kept static along the video sequence after the first frame is learned. When the target has large appearance changes this may lead to tracking failure.
- Targets with uniformly distributed appearance patches, such as the T-shirt with a single color, can lead potentially to tracking failures. In SPT, spatial configuration and basis distribution are used to locate the target center. This type of target will generate a flat voting map and spike histogram, and thus reduce the robustness of SPT. A way to alleviate this potential shortcoming is to use a larger kernel, so that the patches around the target boundary can provide a sufficient contribution to locate the target center.

Scaling: Mean-shift and voting methods can locate the target center very well, but they cannot handle scaling in nature. Without handling scaling, the appearance of patches are not matched to the learned target appearance model. In our method, we assume that the target’s scaling is continuous, therefore we can keep recording the target scale from previous frames and try different scales in the current frame. The scale with the best tracking result (highest score) will be selected as final target object scale.

Rotation: Large rotations or non-rigid transformations will make the learned sparse appearance model and learned voting map invalid, thus generating inaccurate results as shown in the left column of Figure 16. To provide better performance for rotation, there are several methods that can be utilized and will be pursued in future work. The first one is to perform tracking in the affine transformed image frame for different rotation parameters similar to the way we handle the scaling in SPT. The tracking result with the highest score will be treated as the final result. It is based on the assumption that the rotation is continuous.
this method is not efficient since the tracking must be performed multiple times for different rotation parameters.

To add more flexibility to the voting map, an online updated voting map can be concatenated to the original voting map learned in the first frame. In this way, the voting map can adapt to the rotation. The right column of Figure 16 denotes the tracking result with an online updated voting map. It is already shown that the combination of a static and a dynamic updated voting map can generate more robust tracking results, especially for rotation.

As future work for handling rotation and scaling, it is worth to explore the dictionary learning pyramid with various scalings and rotations in the training stage. Affine transformation states can be encoded and retrieved for tracking the target in the same manner as the sparse voting algorithm presented in this paper.

8 Conclusion

In this paper, we have developed and tested a robust and novel tracking algorithm with a static sparse dictionary and dynamic online updated basis distribution, which can adapt to appearance changes and limit the drifting. The target appearance is modeled using a sparse coding histogram based on a learned dictionary with novel selection based dictionary learning method called $K$-Selection. The natural combination of static basis and dynamic basis distribution provides a more robust result. The novel sparse representation based voting map and sparse constraint regularized mean-shift together contribute to the good performance. Experimental results compared to the most recent literature demonstrates the effectiveness of the SPT method. The novel methods described in this paper, such as $K$-Selection, voting and sparse constraint regularized mean-shift, could be extended to other computer vision applications.

Acknowledgement: The authors would like to thank for the great help and valuable suggestions provided by the anonymous reviewers.

References

Fig. 11. The tracking results of selected frames on Box sequence using the proposed method (SPT), Simple Tracker (PROST), Multiple Instance Learning (MIL) and Fragment based Tracker (FragTrack).

Fig. 12. The quantitative tracking results on PROST sequences using the proposed method (SPT), Multiple Instance Learning (MIL), Simple Tracker (PROST), and Fragment based Tracker (FragTrack).
Fig. 13. The tracking results of selected frames using the proposed method (SPT), Multiple Instance Learning (MIL), Simple Traker (PROST) and Fragment based Tracker (FragTrack).

Fig. 14. The tracking results of SPT on the celebrity face data [25]. The object is labeled with red rectangle.
(a) The tracking results of the animal sequence with SPT

(b) The tracking results of the football sequence with SPT

(c) The tracking results of the basketball sequence with SPT

Fig. 15. The tracking results of SPT on three sequences with large dynamic motions

Baiyang Liu received his B.E degree in Computer Science and Technology, Shantou University, China in 2006. He was a Ph. D candidate in the Department of Computer Science, Rutgers, The State University of New Jersey USA since 2006. His major research interests focused on machine learning, pattern recognition, medical image analysis and computer vision. Baiyang Liu is with Amazon.com now.

Junzhou Huang received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China in 1996; the M.S. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China in 2003; and the Ph.D. degree from Rutgers University, New Brunswick, NJ, USA in 2011. He is an Assistant Professor in the Computer Science and Engineering department at the University of Texas at Arlington. His research interests include biomedical imaging, machine learning and computer vision, with focus on the development of sparse modeling, imaging and learning for large scale inverse problems.

Casimir Kulikowski is Board of Governors Professor of Computer Science at Rutgers The State University of New Jersey. His research is on pattern recognition, clustering, and knowledge representation, biomedical informatics, computer visualization and imaging, and the historical and societal impact of computers and informatics. Professor Kulikowski received a Bachelor of Engineering in 1965 and a Master of Science degree in Engineering and Applied Science in 1966, both from Yale University. He received a PhD from the University of Hawaii in 1970 on subspace pattern recognition methods for medical diagnosis. He is a member of the Institute of Medicine of the National Academy of Sciences (iOM-NAS) of the United States, a Founding Fellow of the American Academy of Medical Informatics (ACMI) and the American Association of Artificial Intelligence (AAA), and a Fellow of the American Association for the Advancement of Science (AAAS), and the American Institute for Medical and Biological Engineering (AIMBE). He is a Vice-President of the International Medical Informatics Association (IMIA).

Lin Yang is an assistant professor with the Division of Biomedical Informatics at Dept of Biostatistics, and Dept. of Computer Science at University of Kentucky. He received his B. E. and M. S. from Xian Jiaotong University in 1999 and 2002, and his Ph. D. in Dept. of Electrical and Computer Engineering from Rutgers, the State University of New Jersey in 2009. He was an assistant professor in the Dept. of Radiology in University of Medicine and Dentistry of New Jersey from 2009-2011. His major research interests are focused on biomedical image analysis, computer vision, machine learning and imaging informatics. He is also working on high performance computing and computer aided diagnostics.