Robust Tracking Using Local Sparse Appearance Model and K-Selection

Baiyang Liu[1*] Junzhou Huang[1]
[1]Rutgers, The State University of New Jersey
Piscataway, NJ, 08854
baiyang, jzhuang, kulikows@cs.rutgers.edu

Lin Yang[2] Casimir Kulikowsk[1]
[2]UMDNJ-Robert Wood Johnson Medical School
Piscataway, NJ, 08854
yangli@umdnj.edu

Abstract

Online learned tracking is widely used for it's adaptive ability to handle appearance changes. However, it introduces potential drifting problems due to the accumulation of errors during the self-updating, especially for the occluded scenarios. The recent literature demonstrates that appropriate combinations of trackers can help balance stability and flexibility requirements. We have developed a robust tracking algorithm using a local sparse appearance model (SPT). A static sparse dictionary and a dynamically online updated basis distribution model the target appearance. A novel sparse representation-based voting map and sparse constraint regularized mean-shift support the robust object tracking. Besides these contributions, we also introduce a new 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.

1. Introduction

Generative and discriminative methods are two major categories used in current tracking techniques. The generative models formulate the tracking problem as searching for the regions with the highest likelihood [5, 7, 16, 18, 19, 25, 28, 15], either using a single appearance model or multiple appearance models [27]. Discriminative methods formulate tracking as a classification problem [3, 4, 6, 11]. The trained classifier is used to discriminate the target from the background and can be updated online. Grabner et. al. [9] proposed to update the selected features incrementally using the current tracking result, which may lead to potential drifting because of accumulated errors, aggravated by occlusions.

In order to handle the drifting problem, semi-online boosting [10] 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 drifting problem is handled in this method since the true target included in the positive bag is learned implicitly. Recently, it has been shown that an appropriate combination of complementary tracking algorithms can help alleviate drifting problems [12, 26, 21, 20]. In [17, 13], a sparse representation models the dynamic appearance of the target and handles the occlusion as a sparse noise component. All these methods model the target as a single entity, and therefore cannot handle partial occlusion very well. Fragment-based tracking 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 the present paper, we propose and test 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 basis distri-

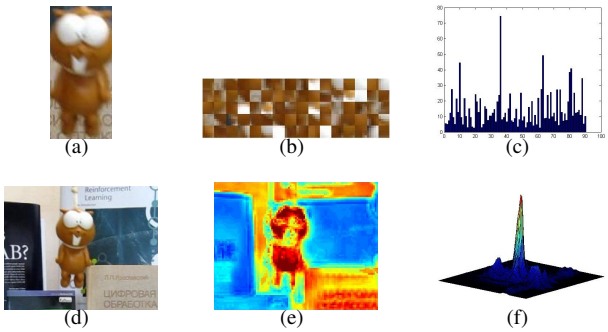


Figure 1. The target appearance (a) is modeled with a dictionary (b) and a sparse coding histogram (c). The confidence map (e) of the image (d) is the inverse of the reconstruction error from the learned target dictionary. The target center is found by voting and sparse constraint with regularized mean-shift on the probability map (f).

*This research is completed when the author is a research assistant in the UMDNJ-Robert Wood Johnson Medicial School

bution which is represented by a sparse coding histogram and is updated online; a sparse representation based voting map and reconstruction error regularized mean-shift which are used to finally locate the center of the object. Besides all these contributions, a novel sparse dictionary learning method, called K-selection, is introduced to learn the target sparse representation library. Figure 1 illustrates an overview of the proposed algorithm. 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 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.

The paper is organized as follows: The local sparse appearance modeling is introduced in Section 2; In Section 3, we describe the new sparse dictionary learning method, K-selection; The tracking procedure is presented in Section 4; Section 5 presents the experiments and section 6 concludes the paper.

2. Local Sparse Appearance Model

Sparse representation has been widely used in many fields including visual tracking [17, 13]. With the sparse assumption, the given signal x can be represented as the linear combination of a few basis vectors in the collected library Φ with component f representing the noise:

$$x = \Phi\alpha + f. \tag{1}$$

The representation coefficient α can be computed by optimizing the l_1 regularized least square problem, which typically provides a sparse solution [24]:

$$\alpha^* = argmin_\alpha ||x - \Phi\alpha - f||^2 + \lambda||\alpha'||_1 \tag{2}$$

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

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.

2.1. Locally 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 ith column vectorized image patch. If $p = m \times n$ is its dimension, and $\Phi \in R^{p\times K}$ is the basis dictionary learned from the target patch set, the target patches can be reconstructed by solving Eqn. (2). In tracking applications, since similarity is more essential than sparsity, Locality-constrained Linear Coding (LLC) [23] is utilized in our algorithm. In LLC, the objective function in Eqn. (2) is changed to:

$$\begin{aligned} &\min_\alpha ||x - \sum_{i=1}^{K} \Phi_i\alpha_i||^2 + \lambda||d \odot \alpha||^2, \\ &s.t. 1^T\alpha = 1 \end{aligned} \tag{3}$$

where $\odot$ denotes element-wise multiplication and d is the Euclidean distance vector between y and all basis vectors in Φ. Constraint $1^T c = 1$ ensures the shift-invariance. The solution α is not l_0 norm sparse, but has only a few significant components.

The second term in Eqn. (3)penalizes the distance from the candidate patch to the basis. LLC actually selects a set of local basis vector 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 [23].

$$\begin{aligned} &\min_{\hat{\alpha}} ||x - B\hat{\alpha}||^2, \\ &s.t. 1^T\hat{\alpha} = 1 \end{aligned} \tag{4}$$

The sparse solution of Eqn. (3) can be approximated by projecting back from $\hat{\alpha}$. In terms of tracking, LLC is used to compute the representation with the local template patches (basis) in Φ that are similar to the candidate sample x.

2.2. Sparse Coding Histogram

Since the target appearance is modeled by local patches, global structure information is necessary for accurately identifying the target. Here we define the sparse coding histogram representing the basis distribution for both target model and target candidates.

Target Model: 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 pixels far away from the center. The value of the j-th bin, q_j, in the target model can be computed as :

$$q_j = C\sum_{i=1}^{N} k(||c_i||^2)|\alpha_{ij}| \tag{5}$$

where C is a normalization constant to ensure $\sum_{j=1}^{K} q_j = 1$, and α_{ij} is the j-th coefficient of the i-th image patch.

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 :

$$\hat{p}_j(y) = C\sum_{i=1}^{N'} k(||\frac{y - c_i}{h}||^2)|\alpha_{ij}^*| \tag{6}$$

where α^* 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 $\hat{p}_j(y)$ its coding histogram from Eqn. (6), then the new appearance basis histogram can be updated with learning rate γ:

$$q'_j = q_j(1-\gamma) + \hat{p}_j(y)\gamma \tag{7}$$

3. Dictionary Learning by K-Selection

We assumed that the dictionary Φ is given in all the above. Many methods have been proposed to learn a dictionary for minimizing the overall reconstruction error for sparse representations [8, 14, 2]. The target templates are stored and updated to form the dynamic dictionary in [17, 13]. Here, we proposed a new method for learning the dictionary as basis selection by gradient descent. Given a dataset $X = \{x_i | i = 1...N\}$, the problem can be formulated as selecting K data vectors as a basis from the dataset which can minimize the objective function:

$$f(\Phi) = \sum_{i=1}^{N} ||x_i - \sum_{k=1}^{K} \Phi_k \alpha_{ik}||^2 + \lambda ||d_i \odot \alpha_i||^2,$$
$$s.t. 1^T \alpha_i = 1, \forall i. \tag{8}$$

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 and α_i are the x_i's distance to the dictionary and the representation coefficients. Exhaustive search could be needed to find the optimal solution, so we propose an efficient method which can converge to a suboptimal solution, called K-Selection.

3.1. 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

$$min_{\omega_i} ||x_i - \sum_{j=1, j\neq i}^{N} x_j \omega_{ij}||^2 + \lambda ||d_i \odot \omega_i||^2,$$
$$s.t. 1^T \omega_i = 1, \tag{9}$$

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

$$w_j = \sum_{i=1}^{N} |\omega_{ij}| e^{-\epsilon_i^2/\sigma^2}. \tag{10}$$

The reconstruction error ϵ_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.

3.2. Gradient Descent

After initialization, a new data vector will be selected to replace the t-th basis to minimize the cost function iteratively. Let α_i be the LLC for x_i with the current Φ 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

$$\nabla f_t = \partial f / \partial \Phi_t = -2 \sum_{i=1}^{N} (x_i - \sum_{k=1}^{K} \Phi_k \alpha_{ik}) \alpha_{it}. \tag{11}$$

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

$$COR(x_l, x_{b_t}, r_t) = \frac{(x_l - x_{b_t})^T r_t}{||(x_l - x_{b_t})||_2 ||r_t||_2}. \tag{12}$$

The data point x_l^* 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} the residual after replacing the t-th basis with x_l^*, 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 5.1.

4. Tracking

4.1. 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 likelihood of all patches within W,

$$P(y|\Phi) = C \prod_{i=1}^{N} e^{-k(||\frac{y-c_i}{h}||^2) \frac{\epsilon_i^2}{\sigma^2}}. \tag{13}$$

where ϵ_i is the sparse reconstruction error for the i-th patch.

The log likelihood of the target candidate is then:

$$L(y|\Phi) = \sum_{i=1}^{N} -k(||\frac{y-c_i}{h}||^2)\frac{\epsilon_i^2}{\sigma^2}. \quad (14)$$

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

$$d(y) = \sqrt{1-\rho(\hat{p}(y),q)}, \quad (15)$$

$$\rho(\hat{p}(y),q) = \sum_{j=1}^{K}\sqrt{\hat{p}_j(y)q_j}. \quad (16)$$

The target then can be located by maximizing:

$$\hat{\rho}(y,\Phi) = \sum_{j=1}^{K}\sqrt{\hat{p}_j(y)q_j}L(y|\Phi). \quad (17)$$

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 Φ.

Assume we have an initial guess of the center as y_0. By Taylor expansion, Eqn. (17) can be rewritten as:

$$\begin{aligned}\hat{\rho}(y,\Phi) &\approx -\frac{1}{2}\sum_{j=1}^{K}\sqrt{\hat{p}_j(y_0)q_j}L(y_0|\Phi) \\ &+ \sum_{j=1}^{K}\sqrt{\hat{p}_j(y_0)q_j}L(y|\Phi) \\ &+ \frac{1}{2}\sum_{j=1}^{K}\hat{p}_j(y)\sqrt{\frac{q_j}{\hat{p}_j(y_0)}}L(y_0|\Phi) \\ &= C_1 + \frac{1}{2}\sum_{i=1}^{N} w_i k(||\frac{y-c_i}{h}||^2), \quad (18)\end{aligned}$$

$$w_i = \sum_{j=1}^{K}\sqrt{\hat{p}_j(y_0)q_j}\left(\frac{L(y_0|\Phi)|\alpha_{ij}|}{\hat{p}_j(y_0)} + \frac{-2\epsilon_i^2}{\sigma^2}\right) \quad (19)$$

where C_1 in $\hat{\rho}(y,\Phi)$ does not depend on y. The second term in Eqn. (18) has to be maximized to minimize the Bhattacharyya distance. It also represents the density estimation computed with kernel $k(.)$ at y with weight w_i. The target center can be found iteratively using Mean-shift.

$$\hat{y}_1 = \frac{\sum_{j=1}^{N'} c_i w_i g(||\frac{\hat{y}_0-c_i}{h}||^2)}{\sum_{j=1}^{N'} w_i g(||\frac{\hat{y}_0-c_i}{h}||^2)} \quad (20)$$

To measure the size of the target, the tracking procedure can be carried out with several scale values and the target center and scale with the maximum of Eqn. 17 selected as the tracking result. The motion and scaling are assumed to be continuous.

Algorithm 1: Compute sparse representation based voting
Define: x_i as the ith image patches centered at position c_i.

1. initialize $V = 0$.
2. for $i = 1 : N$
3. $\quad \alpha_i^* \leftarrow$ solution of LLC Eqn. (3)
4. $\quad \epsilon_i = ||x_i - \Phi\alpha_i^*||_2$;
5. $\quad$ for $j = 1 : K$
6. $\quad\quad V(c^*) = V(c^*) + P(c^* - c_i, j)(1-\delta(\alpha_{ij}))e^{-\epsilon_i^2/\sigma^2}$
7. $\quad$ end
8. end

4.2. Voting in a Sparse Representation

For accurately locating the target center in an occlusion scenario, a voting map is used to improve the robustness of our tracker. In the training stage, not only is the appearance of the image patch, but also the *spatial configuration* in relation to the target center as origin, is encoded in the learned sparse target library. Denote the target center as c_0, for a training patch at position c_i, its spatial information is recorded in the j-th basis as:

$$P_c(d_c, j) = P_c(d_c, j) + \alpha_{ij}^2 k(||d_c/h||^2) \quad (21)$$

where $d_c = c_0 - c_i$ is the offset of the target center relative to the training patch.

In the testing stage, denote x_i as the i-th image patch centered at c_i and α_i^* as its coefficients calculated by LLC. The overall target center voting score is:

$$V(c) = \sum_{i=1}^{N}\sum_{j=1}^{K} P_c(c - c_i, j)(1 - \delta(\alpha_{ij}))e^{-\epsilon_i^2/\sigma^2} \quad (22)$$

The $e^{-\epsilon_i^2/\sigma^2}$ in Eqn. (22) weighs 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 testing stage are given in Algorithm 1. Using the voting map, the final tracking result can be found iteratively with

$$\hat{y}_1 = \frac{\sum_{j=1}^{N'} c_i w_i V(c_i) g(||\frac{\hat{y}_0-c_i}{h}||^2)}{\sum_{j=1}^{N'} w_i V(c_i) g(||\frac{\hat{y}_0-c_i}{h}||^2)} \quad (23)$$

5. Experiment

In this section, we evaluate our sparse tracking algorithm (SPT) on eight challenging sequences and compare its performance with five other latest state-of-the-art trackers . For the comparison, either the binaries or source codes provided by the authors with the same initialization and parameter settings were used to generate the comparative results. The first three sequences (David, girl, car), the fourth

sequence (faceocc2) and other sequences (lemming, box, board, liquor) can be downloaded from the URLs [1] [2] [3] respectively. The challenges of these sequences are summarized in Table 1, including pose variation, illumination changes, occlusions and scaling. Overall, our SPT method provides more accurate and stable tracking results. The videos and codes are provided on the project website[4].

Table 1. The challenges of the experimental sequences

Sequence	3D Pose	Illumination	Occlusion	Scaling
David	√‡	√†	×	√
girl	√‡	√	√‡	√
car	×	√‡	×	√
faceocc2	×	√†	√‡	×
lemming	√	×	√‡	√
box	√	√	√‡	√
board	√‡	×	√	√
liquor	√	√	√†	√

‡ Heavy variation or occlusion.
† Partial illumination changes or occlusion.

5.1. Reconstruction vs. Discriminative Power

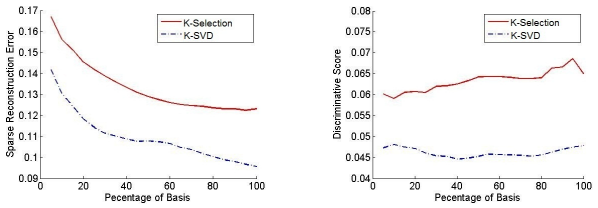
As noted in Section 3, it is worthwhile to evaluate not only the reconstruction error, but also the discriminative power for a sparse dictionary learning method. We claim that under certain conditions, discriminative power weighs more than actual reconstruction error. In this set of experiments, more 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

$$E(X) = \frac{1}{N}\sum_{i=1}^{N} ||x_i - \Phi\alpha_i^*|| \tag{24}$$

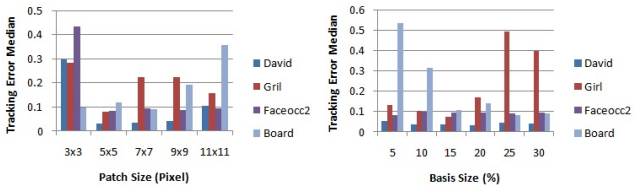
where α_i^* is the sparse solution of the i-th patch using Eqn. (3). The difference between $E(X^+)$ and $E(X^-)$ is used to measure the discriminative power of the learned dictionary. A larger difference $|E(X^+) - E(X^-)|$ indicates stronger discriminative power.

The popular K-SVD method [2] using Orthogonal Matching Pursuit (OMP) [22] is used for comparison. As shown in Figure 2(a), it is not surprising that the dictionary learned with K-SVD has a smaller reconstruction error than our K-Selection method as it explicitly minimizes the l_2 reconstruction error. However, the dictionary learned by K-SVD with OMP can also represent the background patches,



(a) Reconstruction errors (b) Discriminative powers

Figure 2. Performance of the learned dictionaries using K-Selection (red) and K-SVD (blue dash) evaluated by reconstruction errors (a) and discriminative power (b).



(a) Size of the patches (b) Percentage of the basis

Figure 3. Analysis of the tracking performance with different size of patches (a) and different percentage of the selected basis (b).

which leads to relatively weaker discriminative power compared with our K-Selection algorithm, as shown in Figure 2(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 experimentally found that a very small basis set is frequently sufficient to discriminate target from background.

5.2. Parameter Analysis

Our algorithm has two important parameters: patch size s and percentage β of the selected basis over the whole training set. Four sequences(David, girl, faceocc2, board), exhibiting illumination changes, pose variations and occlusions were tested with $s = (3\times3, 5\times5, 7\times7, 9\times9, 11\times11)$ and $\beta = (5\%, 10\%, 15\%, 20\%, 25\%, 30\%)$.

As we can observed in Figure 3(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 3(b). A larger dictionary will degenerate tracking performance due to the loss of discrimination.

5.3. Comparative Tracking Results

Benchmark Sequences: Our SPT method was first evaluated on four benchmark sequences compared with: Multiple Instance Learning (MIL) [4], Online Simple Track-

[1] http://www.cs.toronto.edu/∼dross/ivt/
[2] http://vision.ucsd.edu/∼bbabenko/project_miltrack.shtml
[3] http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php
[4] http://paul.rutgers.edu/∼baiyang/spt

Table 2. Comparative results on the benchmark datasets.

	david	girl	car	faceocc2
PROST[20]	0.124	0.115	NA	0.116
TST[13]	0.052	0.131	0.065	0.139
MIL[4]	0.127	0.161	0.700	0.095
IVT[19]	0.059	0.147	**0.020**	0.081
SPT	**0.026**	**0.066**	0.031	**0.065**

Table 3. Comparative results on the PROST datasets.

	lemming	box	board	liquor
PROST[20]	0.189	0.091	0.157	0.101
FragTrack[1]	0.625	0.406	0.363	0.145
MIL[4]	0.112	0.740	0.206	0.619
SPT	**0.101**	**0.073**	**0.059**	**0.016**

ing (PROST) [20], Two Stage Sparse Tracker (TST) [13] and Incremental Visual Tracking (IVT) [19]. The quantitative results are shown in Table 2. 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 pixel-wise tracking errors (measured by the Euclidean distance from the center of the target to the ground-truth) are shown in Figure 4. Comparative results for selected frames are presented in Figure 5. Our method produces the smallest tracking error for David, girl, faceocc2 sequences which have large appearance changes. IVT yields slightly better results for car sequence, which has heavy illumination change but smaller appearance changes. MIL is good in general, but experiences difficulty in handling partial illumination change ($\#44$ David). As shown in Figure 4(c), MIL drifted away after heavy illumination changes of the target under a bridge ($\#240$ car). IVT and TST have relatively larger errors in the girl's sequence when the target is occluded by similar faces ($\#424, \#436$ girl). When the target varies due to either illumination changes, pose variations or similar occlusion, our SPT method provides robust and accurate tracking results.

PROST Sequences: To further evaluate our SPT method for accurate tracking under occlusion, appearance blur, and pose variation, the latest four sequences provided in [20] are selected to compare with PROST, MIL, and FragTracker [1]. Our SPT method gave the best performance for all sequences, as shown in Table 3, 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 ($\#336$ lemming, $\#300$ box, $\#731$ liquor). MIL and PROST cannot track the target accurately when large pose variation occurs ($\#994$ lemming, $\#600$ box, $\#497$ board), while our SPT method can track the target even under 90 degree off-plane rotation ($\#497$ board).

6. Conclusion

We have developed and tested a robust 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 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 constraints regularized mean-shift together contribute to the robust performance. Experimental results compared to the most recent literature demonstrates the effectiveness of the SPT method.The algorithms described in this paper, such as K-Selection, voting and sparse constraint regularized mean-shift, could be extended to other computer vision applications.

References

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

[2] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311 –4322, 2006.

[3] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007.

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

[5] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):329–342, 1998.

[6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. *ICCV*, 2009.

[7] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-Based Object Tracking. *PAMI*, 25(5):564–575, April 2003.

[8] K. K. Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Comput.*, 15(2):349–396, 2003.

[9] H. Grabner and H. Bischof. On-line boosting and vision. *CVPR*, 2006.

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

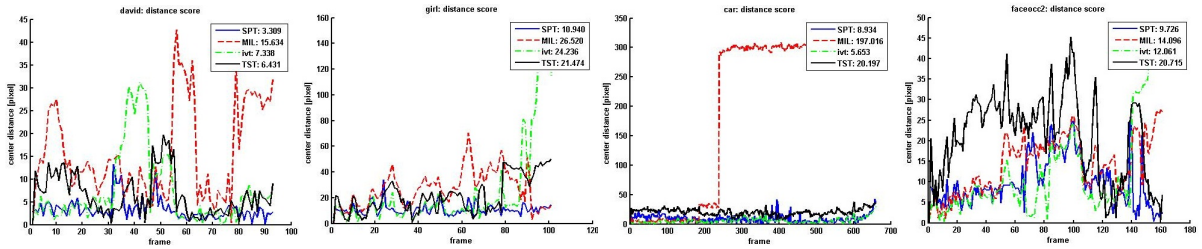
[11] R. Hess and A. Fern. Discriminatively trained particle filters for complex multi-object tracking. *CVPR*, 2009.

[12] I. Leichter, M. Lindenbaum, and E. Rivlin. A probabilistic framework for combining tracking algorithms. *CVPR*, 2004.

[13] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. *ECCV*, 2010.

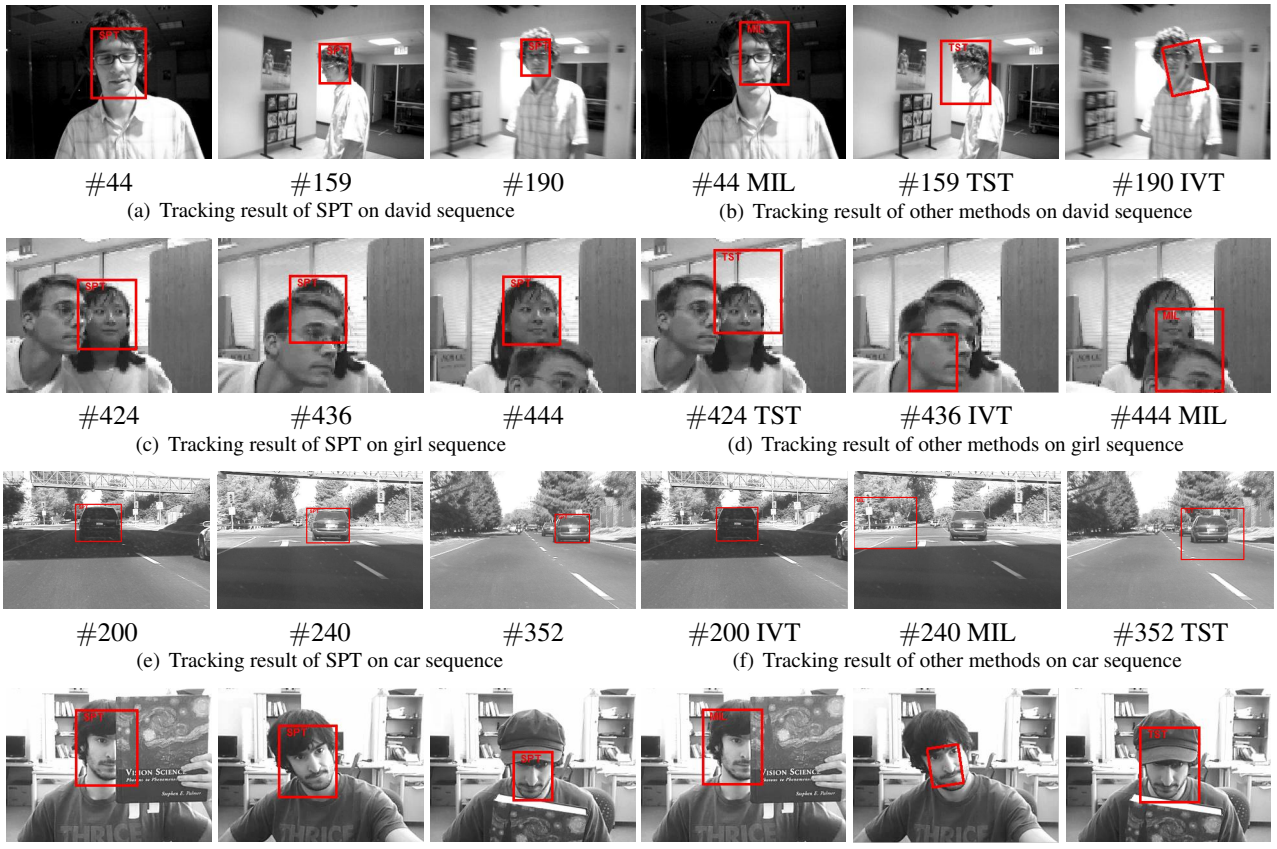
[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. *ICML*, 2009.

[15] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.



(a) david (b) gril (c) car (d) faceocc2

Figure 4. Comparative results on benchmark sequences with our method (SPT), Multiple Instance Learning (MIL), Two Stage Tracker (TST) and Incremental Visutal Tracker (IVT) .



#44 #159 #190 #44 MIL #159 TST #190 IVT

(a) Tracking result of SPT on david sequence (b) Tracking result of other methods on david sequence

#424 #436 #444 #424 TST #436 IVT #444 MIL

(c) Tracking result of SPT on girl sequence (d) Tracking result of other methods on girl sequence

#200 #240 #352 #200 IVT #240 MIL #352 TST

(e) Tracking result of SPT on car sequence (f) Tracking result of other methods on car sequence

#268 #540 #741 #268 MIL #540 IVT #741 TST

(g) Tracking result of SPT on faceocc2 sequence (h) Tracking result of other methods on faceocc2 sequence

Figure 5. Result of selected frames from our method (SPT), Multiple Instance Learning (MIL), Two Stage Tracker (TST) and Incremental Visutal Tracker (IVT) .

[16] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26(6):810–815, 2004.

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

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

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

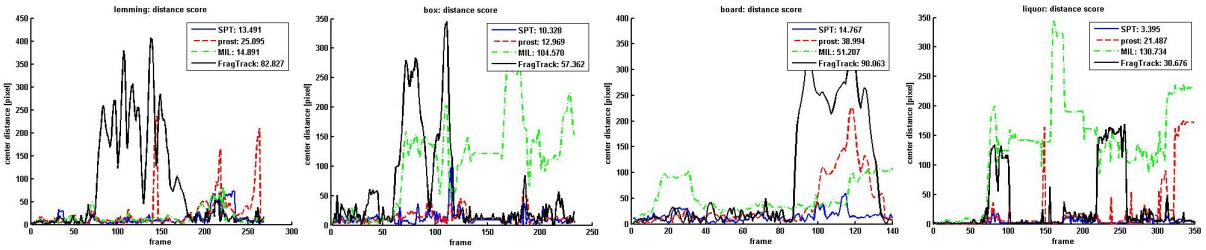
[20] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. *CVPR*, 2010.

[21] B. Stenger, T. Woodley, and R. Cipolla. Learning to track with multiple observers. *CVPR*, 2009.

[22] J. A. Tropp, Anna, and C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transation on Information Theory*, 53:4655–4666, 2007.

[23] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *CVPR*, 2010.

[24] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust

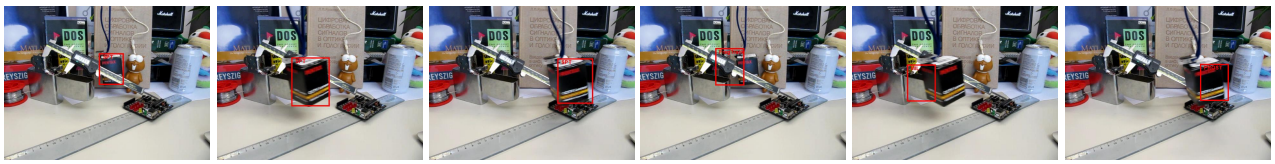


(a) lemming (b) box (c) board (d) liquor

Figure 6. Comparative results on PROST sequences with our method (SPT), Multiple Instance Learning (MIL), Simple Traker(PROST) and Fragment based Tracker (FragTrack) .

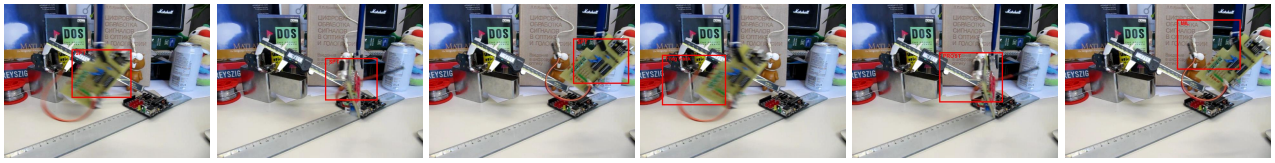
#336 #376 #994 #336 MIL #376 FragTrack #994 PROST

(a) Tracking result of SPT on lemming sequence (b) Tracking result of other methods on lemming sequence



#300 #600 #928 #300 FragTrack #600 MIL #928 PROST

(c) Tracking result of SPT on box sequence (d) Tracking result of other methods on box sequence



#436 #497 #591 #436 FragTrack #497 PROST #591 MIL

(e) Tracking result of SPT on board sequence (f) Tracking result of other methods on board sequence

#355 #451 #731 #355 MIL #451 FragTrack #731 PROST

(g) Tracking result of SPT on liquor sequence (h) Tracking result of other methods on liquor sequence

Figure 7. Result of selected frames from our method (SPT), Multiple Instance Learning (MIL), Simple Traker(PROST) and Fragment based Tracker (FragTrack) .

face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[25] M. Xue, S. K. Zhou, and F. Porikli. Probabilistic visual tracking via robust template matching and incremental subspace update. *ICME*, 2007.

[26] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu. 3D ultrasound tracking of the left ventricles using one-step forward prediction and data fusion of collaborative trackers. *CVPR*, 2008.

[27] Q. Yu, T. B. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. *ECCV*, 2008.

[28] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *TIP*, 13(11):1491–1506, 2004.