Online Object Representation Learning and its Application to Object Tracking

Amirreza Shaban[1] *shaban@ce.sharif.edu*
Hamid R. Rabiee[1] *rabiee@sharif.edu*
Mehrdad Farajtabar[2] *mehrdad@gatech.edu*
Mohsen Fadaee[1] *m_fadaee@ce.sharif.edu*

[1]Department of Computer Engineering,
Sharif University of Technology,
Tehran, Iran.

[2]College of Computing,
Georgia Institute of Technology,
Atlanta, GA, USA.

Abstract

Tracking by detection is the topic of recent research that has received considerable attention in computer vision community. Mainly off-line classification methods have been used, however, they perform weakly in the case of appearance changes. Training the classifier incrementally and in an online manner solves this problem, but nevertheless, raises drifting due to soft or hard labeling in the online adaptation. In this paper a novel semi-supervised online tracking algorithm based on manifold assumption is proposed. Target object and background patches lie near low-dimensional manifolds. This motivates us to make use of the intrinsic structure of data in classification, and benefit from the smooth variation of the labeling function with respect to the underlying manifold. Unlabeled data make connections between different object poses to overcome difficulties due to appearance changes and partial occlusion. Moreover, the proposed method doesn't rely on self-training, therefore, it is more robust to drifting. Experimental results substantiate the superiority of the proposed method over the ones that does not consider the geometry of data.

Introduction

Tracking is the process of pursuing an object in an image sequence. It has a variety of applications in human-computer interaction, security systems, medical imaging. There are a large number of challenges associated with tracking: shape and appearance changes, illumination variations, difficulty in distinguishing the target object from its background due to the background clutter, non-linear dynamics of object appearance caused by pose variations, and partial occlusion. These issues make tracking a hard problem to tackle with.

Traditional algorithms for tracking merely focus on modeling the object, and completely ignore the background (Balan and Black 2006). This approach fails when the background is complicated and the object is not easily distinguishable from its background. To overcome these drawbacks (Lin et al. 2004) and (Avidan 2004) initiate the idea of tracking with a discriminative approach, by considering background as well as the object. This way, tracking is regarded as a background/foreground classification problem. The data to be classified are patches from the image.

Training the classifier in an off-line manner raises an immediate challenge; when the appearance of the target object dramatically changes during tracking, the tracker fails. To cope with this problem, object model should be updated during the tracking, i.e., the decision boundary is learned in an online manner. An online boosting classifier is constructed in (Grabner and Bischof 2006) that acts as a feature selector to discriminate the object from the background. Authors in (Babenko, Yang, and Belongie 2009) cast tracking as a multiple instance learning problem. Although very robust, it suffers from being intrinsically a variation of self-training, the demerits of which will be discussed shortly. Also Avidan et al. (Avidan 2005) adopt an ensemble of SVMs as a classifier for tracking.

Although these methods provide a solution to varying appearance issue, introduce a new problem: the drifting of the classifier. Since the aforementioned online methods are all rooted in self-training, during the tracking, erroneous updates might be made to the algorithm which results in accumulation of errors and tracker failure. To defeat this new problem semi-supervised learning comes to help. Instead of hard labeling the detected objects and its surroundings, they treat them as unlabeled data. To elaborate, they use some kind of pseudo-labeling in order to alleviate the disadvantages of hard-labeling. Inspired by co-training, authors in (Tang et al. 2007) proposed two online SVM classifiers based on independent features which train each other using unlabeled data. In (Grabner, Leistner, and Bischof 2008) authors modified semi-boost (Mallapragada et al. 2009) to operate in an online manner. A novel idea for online semi-supervised tracking is proposed in (Zeisl et al. 2010) where multiple instance learning is used to manage the updating of the classifier during tracking.

Soft-labeling procedure has two disadvantages, First these soft-labels are assigned according to the current knowledge of the data distribution, and are constant in time. As a result, further exploration of the problem space, may lead to invalidation of the primarily assigned pseudo-labels. Invalid labels may then mislead future tracking, especially in initial training levels where the classifier is only moderately accurate (Zhang and Zhou 2010). Second, the classifier may over-focused on the last few frames, so it fails in rapid appearance change of object.

In this paper, a novel semi-supervised tracking algorithm is proposed which learns low dimensional manifolds of object and background in an online manner. Our approach is truly semi-supervised, in the sense that we perform neither hard-labeling nor pseudo-labeling. In graph based semi-supervised methods, maintaining the repository of accumulated object appearances is the main problem. We suggested to omit the redundant data by coarse graining so we can keep main appearances of the object during the tracking and use this knowledge for further usages. Unlabeled data connect the initial labels (which are of high confidence) to the future data in a natural way, thus there is no more need for any kind of labeling. In addition our method does not suffer from over-focusing in one region of the manifold, since it utilizes all the appearances of the object seen so far.

Our contribution is three fold: First, we propose to use classification methods based on manifold assumption in tracking which is supported by the very essence of data. Second, a novel coarse graining algorithm is proposed for data reduction while preserving manifold structure of target and background patches. Third, an incremental method to discriminate object/background patches.

The rest of the paper is organized as follows. First we discuss the advantage of manifold-based classification methods over other tracking methods. Then the proposed method which consists of three steps is thoroughly explained. In the end, experimental results showing the superiority of the algorithm are presented.

Manifold-based Tracking

In many real world learning problems, data points lie on a low dimensional manifold. Visual data strongly support this claim. Image sequences and videos vary with very limited degrees of freedom, therefore, the set of these points lie near a low dimensional manifold. Figure 1 shows patches of a walking person taken from PETS-ECCV 2004 dataset accompanied with some of images. The person starts walking, then stands and raises his hand, and finally walks back. The target object patches are projected into a 2 dimensional space. As can be seen from the evolution of walking, the data don't occupy the whole space, instead, in this 2 dimensional space, the data lie near a 1 dimensional manifold. A good practice is to utilize this prior knowledge (manifold structure) in tracking. The fact that object and background patches lie near low-dimensional manifolds leads to better results in tracking in the same way as manifold learning methods are superior to non-geometric algorithms. The more the data satisfy the manifold assumption the better is the result.

Suppose we have u unlabeled and l labeled points. Let $n = u + l$ be the total number of data points. Also let y_t be a vector of length n with $y_t(i) = 0$ for unlabeled x_i and $y_t(i)$ equals to the -1 or 1 corresponding to the class labels for the labeled data points. Our goal is to predict all the labels, f, where $f(i)$ is the label associated to x_i for $i = 1, \ldots, n$.

A graph is constructed for modeling the manifold. For each data point a node is considered and two nodes are connected if they are locally close enough, i.e., if one of them is among the K nearest neighbors of the other. The weight is usually assigned exponentially parameterized with the bandwidth σ: $W(i,j) = \exp(-\left|\left|x_i - x_j\right|\right|^2 / 2\sigma^2)$. W measures the local similarity of data points. Define the diagonal matrix D with nonzero entries $D(i,i) = \sum_{j=1}^{n} W_{ij}$. Symmetrically normalize W by $S = D^{-1/2} W D^{-1/2}$. The laplacian matrix is $L = I - S$.

Manifold regularization algorithms in their general forms amount to minimize

$$J(f) = f^T Q f + (f - y)^T C (f - y) \qquad (1)$$

with respect to the prediction, f, where Q is a regularization matrix (usually the laplacian itself) and C is a diagonal matrix with $C(i,i)$ equal to the importance of the i^{th} node to stick to its initial value $y_t(i)$. The first term represents smoothness of the predicted labels with respect to the underlying manifold and the second term is squared error of the predicted labels compared with the initial ones weighted by C.

Harnessing local and global consistency(LGC) (Zhou et al. 2004) is one of the pioneering works on manifold-based classification methods. We use it as a base classifier because of its simplicity, robustness against noise and state-of-the-art performance. Semi-supervised tracking uses very small number of labels which can be noisy instances of the object model, thus LGC's robustness against noise makes it very favorable choice for tracking.

In LGC, $Q = L$ and $C = \mu I$. It may easily be shown that the solution is equal to:

$$f = (I - \alpha S)^{-1} y_t, \qquad (2)$$

where $0 < \alpha = \frac{1}{1+\mu} < 1$ is the parameter which controls the trade-off between smoothness of the labeling function on the manifold and the empirical error on the initially labeled points. In subsection we show how to compute LGC's solution in an on-line manner with incrementally computed eigen-pairs of P.

Proposed Method

In this section, we propose a 3-step algorithm which learns the manifold structure of data, summarizes it, and incrementally computes the labels. To address time and memory limitation in online learning, we limit the number of data points which can be kept to, say n, fixed a priori. On the arrival of new data, two data points are selected to be merged in the manner which manifold structure of data is best preserved. It will be discussed shortly that the spectrum of the stochastic matrix helps us to find the two points which are nearest pairs on the manifold in the first step. This spectrum is also maintained in an online manner with the arrival of a new data point in the second step, and help us compute labels of the new data points in the third step. Our algorithm is summarized in Figure 2.

Before proceeding, let us go through some preliminary explanations. Tracking algorithm is converted to a 2-class classification problem by considering object and background patches as positive and negative instances respectively. In semi-supervised learning, there are some labeled

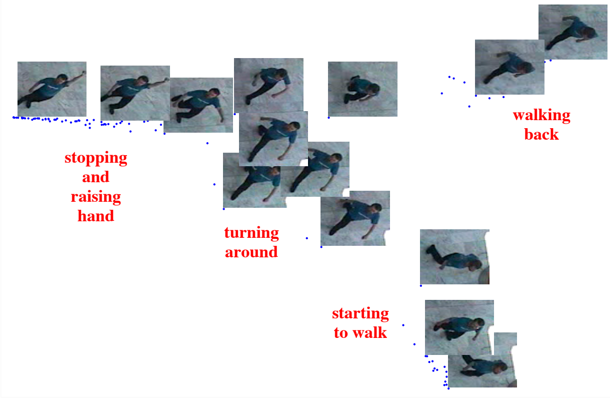

Figure 1: Image patches of a person walking accompanied with some images, which lie near a one dimensional manifold. The Images are from PETS ECCV 2004 dataset.

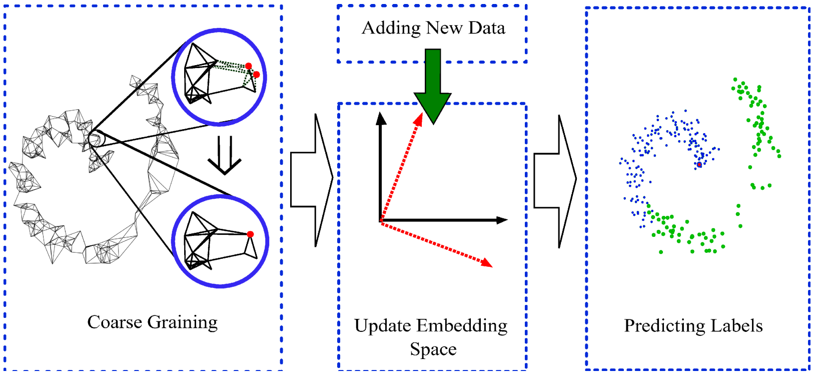

Figure 2: Three steps in the proposed online classification method used in tracking

instances (positive and negative) and some unlabeled data points which are to be labeled in the transductive setting.

For the rest of the paper suppose that the stochastic matrix defined as $P(i,j) = \frac{W(i,j)}{\sum_{j=1}^{n} W(i,j)}$, contains the probability of reaching node j from i. We can write P equivalently as $P = D^{-1}W$.

Since P is a stochastic matrix, its spectrum is real. Let p_i, u_i, and λ_i be its i^{th} right and left eigenvectors and i^{th} eigenvalue respectively, which are sorted decreasingly by eigenvalue. We also denote $P^t(i,j)$ as the probability of reaching j from i in t steps. For P^t which is a stochastic matrix, right and left eigenvectors are the same as those of P and its eigenvalues are those of P powered by t.

Merging on the Manifold

In this section, we provide a method to choose two data points to be merged so that minimum amount of information is lost. By information, we mean all we know about the geometry of data, which is needed to predict labels of new data points. The primary question is which two of these data points are the most similar pair. A good measure should take into account both the manifold structure of data and the distribution of data on it. Assuming the manifold structure of data is captured by the adjacency graph, a good method to exploite distribution of the data in similarity measure is considering all paths connecting two nodes. Inspired by the work of (Lafon and Lee 2006) we employ diffusion distance as a similarity measure. Diffusion distance between two nodes x and z is defined as (Lafon and Lee 2006):

$$L_t^2(x,z) = \sum_y \frac{(P^t(x,y) - P^t(z,y))^2}{u_1(y)}. \qquad (3)$$

$P^t(x,y)$ represents the probability of reaching y from x in t steps. The larger $P^t(x,y)$, the stronger the influence of x on y in the label inference. Using this notion of influence, the distance between two nodes x and z is defined as how these nodes are regarded in the view of other nodes. This quantity is small if the nodes are linked to each other via a lot of short paths, therefore the density is considered. It has been shown that there is a transformation T that embed the data to a space in which Euclidean distance is an approximation of diffusion distance (Lafon and Lee 2006):

$$T : x \longmapsto \begin{bmatrix} \lambda_1^t p_1(x) & \lambda_2^t p_2(x) & \dots & \lambda_k^t p_k(x) \end{bmatrix}^\top \qquad (4)$$

where k is the number of significant eigenvectors in terms of absolute eigenvalues. Influence of other eigenvectors are negligible.

It has been proved in (Lafon and Lee 2006) that minimizing the following term

$$E = \sum_{i} \sum_{x \in S_i} u_1(x) \left|\left|T(x) - C(S_i)\right|\right|^2, \tag{5}$$

with respect to S_i leads to minimum distortion on the transformation T. In the above equation S_i is the i^{th} group formed by merging some nodes and $C(S_i)$ is the center of the group which is defined as average of the points within the group weighted by their normalized degree.

In our specific application we want to merge just two nodes. It is easily shown that choosing the those two ones which are closest to each other in the embedding space minimizes the error bound, E. So the transformation is best preserved. On the other hand, the embedding provided by the eigensystem captures manifold structure. Therefore, in merging process minimum change is occurred on the manifold structure.

Time complexity of computing distances between n k-dimensional vectors and choosing two nearest ones is $O(n^2k)$. Considering n is constant and considerably small this step is quite fast.

Update Embedding Space

On the arrival of a new data point the new stochastic matrix, P_{new}, is easily computed. Then, the embedding which relies on the significant eigenvectors of the stochastic matrix should be updated. A priori knowledge about adding a new data is that the embedding would change a little, therefore eigenvectors distortions is negligible (Kouropteva, Okun, and Pietikäinen 2005). Instead of computing the eigenvectors of P_{new} from scratch, we develop an iterative procedure based on power method which utilizes the previously computed eigenvectors of P_{old}.

Power method calculates the eigenvector with largest eigenvalue of a matrix with a simple iterative procedure starting from a random vector: ($v^{(l)}$ is the estimated eigenvector at step l)

$$v^{(l+1)} = \frac{Pv^{(l)}}{\left|\left|Pv^{(l)}\right|\right|}. \tag{6}$$

It is easily shown that this procedure converges to the largest eigenvector (Saad and Saad 1996). To enhance the convergence speed the starting vector, $v^{(0)}$, is initialized to the eigenvector of P_{old} and padded by one 0 for the element corresponding to the new arrived point.

The eigenvalue is then computed as

$$\lambda = \frac{v^T P v}{v^T v}. \tag{7}$$

After computing the first eigenvector and eigenvalue, p_1 and λ_1, they are extracted out and the same procedure is performed to update the next eigen-pair. Considering the decomposition, $P = \sum_i \lambda_i p_i u_i^T$, subtracting $\lambda_1 p_1 u_1^T$ from P causes the first eigenpair be removed from its spectrum. For every i one can easily see that right and left eigenvectors of the stochastic matrix are related by $u_i(j) = p_i(j) u_1(j)$.

Due to small distortion in eigenpairs, the eigenvectors converges in a few number of iterations (which is practically confirmed) and the algorithm needs a few number of sparse matrix-vector multiplication for each eigenvector which leads to $O(kKn)$, where K is the parameter of K nearest neighbor graph construction and k is the number of eigenvectors based upon we do coarse graining.

Label Prediction

We propose an incremental version of LGC based on incrementally calculated eigen-pairs. We then show how to approximate LGC's solution with the k large eigenvectors of P which are maintained during the algorithm.

To begin with, let (λ, v) is an eigenvector and eigenvalue pair of P, we prove that $((1-\alpha\lambda)^{-1}, D^{1/2}v)$ is an eigenpair of $(I - \alpha S)^{-1}$:

$$\begin{aligned} Pv = \lambda v \quad &\Rightarrow \\ D^{-1}Wv = \lambda v \quad &\Rightarrow \quad D^{-1/2}Wv = \lambda D^{1/2} v \end{aligned} \tag{8}$$

Consider $q = D^{1/2}v$ we have:

$$\begin{aligned} D^{-1/2}WD^{-1/2}q = \lambda q \quad &\Rightarrow \\ (I - \alpha D^{-1/2}WD^{-1/2})q = (1-\alpha\lambda)q & \end{aligned} \tag{9}$$

Therefore q is the eigenvector of $I - \alpha S$ corresponding to eigenvalue $1 - \alpha\lambda$. By inverting $I - \alpha S$ the claim is proved.

If (λ, v) is among the k largest eigenpairs of P then $((1-\alpha\lambda)^{-1}, D^{1/2}v)$ is also one of the k largest eigenvectors of $(I - \alpha S)^{-1}$. Assuming other eigenvalues are not large the LGC's solution can be easily approximated by just these k eigenvectors.

By decomposition we have $(I - \alpha S)^{-1} \approx \Phi \Lambda \Phi^T$, where Φ is the $n \times k$ matrix of significant eigenvectors and Λ is the diagonal matrix of corresponding eigenvalues. Computing $\Phi\Lambda\Phi^T$ and then multiplying by y_t is inefficient; since $\Phi\Lambda\Phi^T$ becomes a non-sparse matrix of n^2 elements. Instead we compute the solution with the priority induced by the parenthesization $(\Phi(\Lambda(\Phi^T y_t)))$ which costs $O(nk^2)$ operations.

The entire tracking algorithm is summarized in the Alg.1.

Experiments

To show effectiveness of our algorithm 4 methods are compared with the proposed one. The first one is Online AdaBoost (OAB) described in (Grabner, Grabner, and Bischof 2006) which is a supervised tracker. Online SemiBoost tracker (OSB) (Grabner, Leistner, and Bischof 2008) and BeyondSemiBoost (BSB) (Stalder, Grabner, and Van Gool 2009) - which are semi-supervised methods - are chosen to compare with our manifold-based tracker. Finally, the fourth one is Frag (Adam, Rivlin, and Shimshoni 2006) which is very robust to partial occlusion.

Combinations of color and edge-based features are used for distinguishing patches. For color features, $\{R, G, B\}$ space is discretized into $4 \times 4 \times 4$ sections and a 64 dimensional histogram is extracted. To compute the HoG feature, the patch is divided into 8×8 sub-patches of equal sizes

Algorithm 1 *Manifold-Based Tracking*

```
Input: Dimention of embedding space $k$, Set of Vertices $\mathcal{V}$, Adjacency Matrix $\mathbf{W}$
Output: Labels for $y$
loop
    $\{$Do this loop for each new patch arrival ($x_{new}$).$\}$
    $(i^*, j^*) \leftarrow \arg\min_{i,j} ||T(x_i) - T(x_j)||$
    $\mathcal{V} \leftarrow (\mathcal{V} - \{x_{i^*}, x_{j^*}\}) \cup \{\frac{x_{i^*}+x_{j^*}}{2}\}$
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{x_{new}\}$ $\{$Adding new arrived patch.$\}$
    Update $\mathbf{W}$
    for $i = 1 \rightarrow k$ do
        $\{$Update Embedding Space$\}$
        repeat
            $v_i^{l+1} \leftarrow \frac{Pv_i^l}{||Pv_i^l||}$
        until convergence
        $\lambda_i \leftarrow \frac{v_i^T P v_i}{v_i^T v_i}$ and $P \leftarrow P - \lambda_i v_i$
    end for
    $\Phi \leftarrow [v_1, \ldots, v_k]$
    $y \leftarrow (\Phi\Lambda(\Phi^T y_t))$ $\{$Label propagation$\}$
end loop
```

. A HoG with 9 orientation bins is computed for each subpatch and the final feature vector is a concatenation of these 9-dimensional vectors.

In practice, there is no need to compute the average of the two selected nodes in coarsening process. Patches to be merged are similar enough for it to be sufficient to consider only one of them as a result of merging. Number of nearest neighbors, K is set to 5. For our datasets buffer size of 300 is seen to be empirically sufficient in order to preserve the geometry of patches. σ is set in the way that edge weights are distributed uniformly from zero to one. In our semi-supervised setting we use only 16 labeled examples. The first frame is used for initial labeling. The user determines the object's bounding box, which is treated as the positive example. 8 other boxes around the target's bonding box are treated as 8 negative examples. In order to balance the positive and negative examples, we suppose 7 predicted targets on the next 7 frames as positive examples. Note that this is just a minor semi-supervised initialization and do not mean a self-training which usually exists in other methods. Because it is just on the first 8 frames of the image sequence.

Manifold Preservation

In the first experiment, we show the effectiveness of our coarse-graining algorithm in summarizing essential data to maintain the manifold. The face of the tiger in Tiger dataset is considered as target to be tracked. In this scenario the buffer size is 200. The 48 most relevant images remaining after a long run of the algorithm are demonstrated in Figure 3. Different poses of the tiger, partially occluded faces and blurred ones are all included in this small number of samples. This number would have been much larger (approximately 1000) without coarse-graining; since a notable amount of redundant patches would have been needed to memorize all the appearance changes.

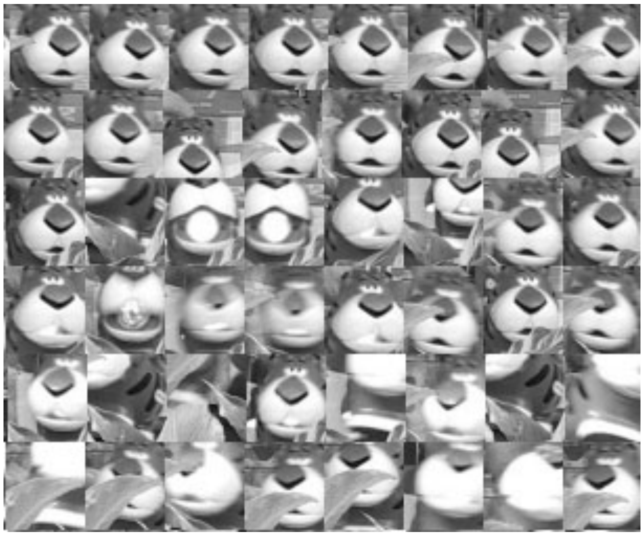

Figure 3: 48 top relevant object patches from the Tiger dataset for tracking tiger's face. Note the different poses and appearances that remain in the set, due to elimination of the redundancy by coarse-graining

Challenges

In this section, we provide some subjective clues to show the effectiveness of the proposed method. The first video is Sylvester and is illustrated in the first row of Figure 4. It is noticeable that our method is robust to appearance changes due to pose and illumination variations. Unlabeled data provided by the manifold of patches help the algorithm connect these appearance changes to the true initial labeled patches. The paths between initial patches and newly arrived patch - which are constructed by the unlabeled nodes of the adjacency graph- make this connection.

The second scenario in the middle row of Figure 4 demonstrates the strength of the proposed method to overcome partial occlusions and appearance changes. Again the manifold is responsible to detect the true patches. Since the partially occluded targets are accessible by paths on the manifold, the proposed method could easily recognize the target.

The third scenario in the last row of Figure 4 shows efficiency of the proposed method in rotation, scaling, and in general appearance changes.

The average center location error (Babenko, Yang, and Belongie 2009) of the proposed method is compared with other methods in Table 1. The bold face and underlined numbers indicate the best and the second best performance, respectively. Our proposed method is always among the two best methods. It is either the best method or the second best provided with a little error compared to the best one.

Conclusion and Future Works

In this paper a novel online semi-supervised tracking algorithm is proposed based on manifold assumption. We claim that by learning the hole manifold the classifier do better in rapid appearance changes. Our tracking algorithm is quite efficient, since no more than a fixed number of data are there

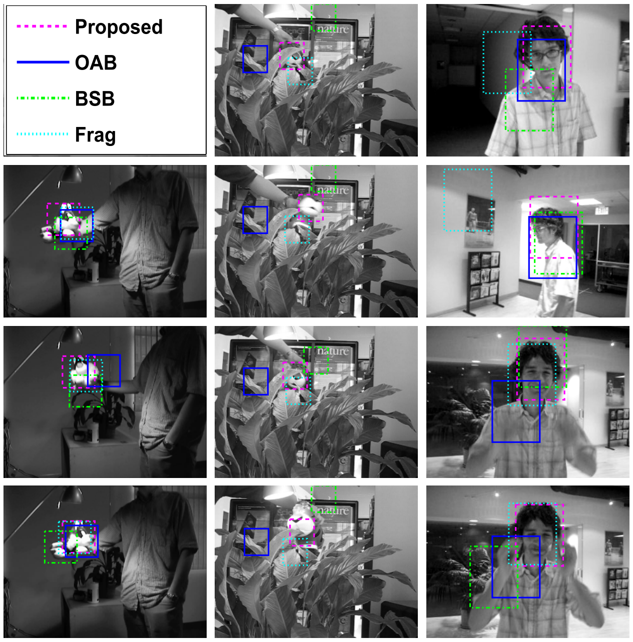

Figure 4: **First column**: (*Sylveser*) Robustness to appearance changes due to pose and illumination variations.
Second column: (*Tiger1*) Overcoming partial occlusions, appearance changes, and background clutter. **Third column**: (*David*) Efficiency in rotation, scaling, and general appearance changes.

Video Clip	OAB	BSB	Frag	OSB	Proposed
Sylvester	17	19	**13**	18	14
Tiger 1	56	33	42	31	**26**
Tiger 2	**23**	42	39	71	25
David	60	58	48	45	**22**

Table 1: Average center location errors for 4 public datasets, compared to 4 other methods.

to be processed. Our method has $O(kn^2)$ time complexity, where n is the buffer size and is problem specific. In our experiments on real-world videos, $n = 300$ was sufficient to capture all the appearance changes of the target object and its background patches. This low time complexity leads to a fast method which is practical interest in real time application. Our coarse graining method decides what instances of object appearance and background patches to save in the buffer in order to maintain the manifold structure of data.

Background patches usually lie on a complicated low dimensional manifold. Better results would be achieved via modeling of background as multiple (and possibly overlapping) manifolds. Each general background entity (road, trees, cars) may separately lie near a manifold. Employing multi-manifold learning algorithms in tracking is another suggested future research trend.

References

Adam, A.; Rivlin, E.; and Shimshoni, I. 2006. Robust fragments-based tracking using the integral histogram. In *CVPR*, volume 1, 798–805. IEEE.

Avidan, S. 2004. Support vector tracking. *PAMI* 26(8):1064–1072.

Avidan, S. 2005. Ensemble tracking. In *CVPR*, volume 2, 494–501. IEEE.

Babenko, B.; Yang, M.; and Belongie, S. 2009. Visual tracking with online multiple instance learning. In *CVPR*, 983–990. IEEE.

Balan, A., and Black, M. 2006. An adaptive appearance model approach for model-based articulated object tracking. In *CVPR*, volume 1, 758–765. IEEE.

Grabner, H., and Bischof, H. 2006. On-line boosting and vision. In *CVPR*, volume 1, 260–267. IEEE.

Grabner, H.; Grabner, M.; and Bischof, H. 2006. Real-time tracking via on-line boosting. In *Proc. BMVC*, volume 1, 47–56.

Grabner, H.; Leistner, C.; and Bischof, H. 2008. Semi-supervised on-line boosting for robust tracking. *Computer Vision–ECCV 2008* 234–247.

Kouropteva, O.; Okun, O.; and Pietikäinen, M. 2005. Incremental locally linear embedding algorithm. *Image Analysis* 145–159.

Lafon, S., and Lee, A. 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *PAMI* 1393–1403.

Lin, R.; Ross, D.; Lim, J.; and Yang, M. 2004. Adaptive discriminative generative model and its applications. *NIPS* 17:801–808.

Mallapragada, P.; Jin, R.; Jain, A.; and Liu, Y. 2009. Semiboost: Boosting for semi-supervised learning. *PAMI* 31(11):2000–2014.

Saad, Y., and Saad, Y. 1996. *Iterative methods for sparse linear systems*. PWS Pub. Co.

Stalder, S.; Grabner, H.; and Van Gool, L. 2009. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *ICCV Workshops*, 1409–1416. IEEE.

Tang, F.; Brennan, S.; Zhao, Q.; and Tao, H. 2007. Co-tracking using semi-supervised support vector machines. In *ICCV*, 1–8. IEEE.

Zeisl, B.; Leistner, C.; Saffari, A.; and Bischof, H. 2010. On-line semi-supervised multiple-instance boosting. In *CVPR*, 1879–1879. IEEE.

Zhang, M., and Zhou, Z. 2010. Exploiting unlabeled data to enhance ensemble diversity. In *ICDM*, 619–628. IEEE.

Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. *NIPS* 16:321–328.