Fast Inference of Contaminated Data for Real Time Object Tracking

Hao Zhu[1], Yi Li[2]

[1] 3M Cogent Beijing R&D Center, Beijing, China
[2] NICTA and Australian National University, Canberra, Australia
ahzhu@mmm.com, yi.li@nicta.com.au

Abstract. The online object tracking is a challenging problem because any useful approach must handle various nuisances including illumination changes and occlusions. Though a lot of work focus on observation models by employing sophisticated approaches for contaminated data, they commonly assume that the samples for updating observation model are uncorrupted or can be restored in updating. For instance, in particle filter based approaches every particle has to be restored for each frame, which is time-consuming and unstable. In this paper, we propose a novel scheme to decouple the observation model and its update in a particle filtering framework. Our efficient observation model is used to effectively select the most similar candidate from all particles only, by analyzing the principal component analysis (PCA) reconstruction with L_1 regularization. In order to handle the contaminated samples while updating observation model, we adopt on an online robust PCA during the update of observation model. Our qualitative and quantitative evaluations on challenging dataset demonstrate that the proposed scheme is competitive to several sophisticated state of the art methods, and it is much faster.

1 Introduction

Visual tracking has been an active topic in computer vision because it is widely used in many applications such as surveillance, robotics, human computer interaction, vehicle tracking, and even medical imaging. In spite of great progress in last two decades, visual tracking is still an extremely challenging topic because in the real scenes visual tracker has to face different situations (e.g. sophisticated object shape or complex motion, illumination changes and occlusions).

The current methods of visual tracking can be categorized into generative or discriminative ones. The methods based on generative models aim at finding the most similar region as the target from a lot of candidates, while the methods based on the discriminative models are modeling the tracking problem as a classification problem, which build classifiers for distinguishing the target from the surrounding region of backgrounds. In this paper, we mainly focus on the visual tracking based on generative models.

Among the trackers based on generative methods, the linear representation is widely employed in many trackers because it is capable of maintaining holistic

appearance information and casting generative models (i.e. Gaussian model) as linear regression. These methods often adopt a dictionary (e.g, a set of basis vectors from a subspace or a series of templates) to represent the tracked target. A given candidate sample is linearly represented by the dictionary, and the representation coefficient and reconstruction error are computed, from which the corresponding likelihood (the similarity to the expected target) is determined.

Ross et al. proposed an incremental visual tracking(IVT) method [1] which employs a low dimensional PCA subspace to represent the tracked target and thus assumes that the error is Gaussian distributed with small variances (i.e., small and dense noise). Technically it is equivalent to the ordinary least squares solution under the assumption that the dictionary atoms are orthogonal, and the representation of the tracked target is capable of obtaining by inner product operator. Furthermore, the reconstruction error is also easy to compute. Thus, the IVT is a potential method for real time applications. While the IVT method is able to handle appearance changes caused by illumination variation and pose variation, it is not robust to some challenging environments (e.g. partial occlusion and background clutter) due to the following two reasons. Firstly, the ordinary least squares method has been shown to be sensitive to outliers due to the formulation is equivalent to Maximum Likelihood Estimate(MLE) under Gaussian models. Secondly, the IVT method directly uses new observations to update the observation model without any intervention such as detecting outliers and processing them accordingly.

Recently, sparse representation has been successfully employed in classification problems in computer vision. Wright et al. [2] reported sparse representation for face recognition (FR). Such a sparse representation classifier (SRC) firstly codes the query face image over the dictionary sparsely, and then makes the classification by checking which class yields the least coding error. This scheme presents an impressive performance in FR due to the robustness to different situations (e.g. face expression, illumination changes and occlusion).

Inspired by SRC, many L_1 trackers represent a candidate target by a sparse linear combination of the templates in a dictionary. Benefitting from sparsity penalty(i.e. L_0 norm and L_1 norm), these methods demonstrated robustness in various tracking environments. However, sparsity penalty is a non-differential function. Therefore these SRC based trackers are quite computationally expensive. In the classical L_1 tracker [3], L_1 minimization problems need to be solved by the interior point method [4] for each frame during the tracking process. This process is very time-consuming not only because L_1 minimization is computationally expensive but also the trivial templates significantly extend the size of the dictionary. The optimized solution proposed in [5] reduces the number of particles by a minimal error bounding strategy. Though this strategy is able to save 80% computation. It is still far away from real time applications. Furthermore, the L_1 based trackers commonly select down-sampling particles as templates due to computational burdens [6], which significantly reduces the tracking accuracy.

Aforementioned L_1 based trackers only concern with the robust representation for observing data and employing raw inputs as "templates", which may

be corrupted or contaminated. These methods ignore that corrupted templates will have significant influences on the observation model because the templates in the dictionary came from raw observations. Unlike the methods that try to solve these two problems by using sophisticated appearance models and better mechanisms of updating model, in this paper we ask two different yet essential questions. 1) Is it necessary to use sophisticated observation methods to handle corrupted data, such as partial occlusion situation? For example, it is very computationally expensive if we have to restore hundreds of particles in each frame) but only need one of them to update the observation model. 2) Is there an efficient and effective way to update the observation model, without occlusion detection in the appearance model?

In this paper, we propose to overcome the disadvantage of subspace representation by proposing an robust appearance model to deal with heavy occlusion effectively. We chose the observation model in the IVT to our model, because IVT performs remarkbly more efficient than L_1 tracker in handling higher resolution image observations. During the model update, we propose to use robust online dictionary learning based method, such as those methods based on Huber loss function (Wang and Yeung [7]). which remedy the problem of corrupted samples by obtaining more robust templates. Several experiments on challenging video sequences validate that the proposed algorithm is efficient and effective for object-tracking problem.

2 Related work

To facilitate the comparison between different methods, we briefly review the particle filter models for visual tracking and trackers based on linear representation. And then some classical trackers based on linear representation (e.g. IVT and L1 Tracker) are also reviewed briefly.

2.1 Particle filter tracking

In the framework of particle filtering, the problem of object tracking can be considered as a sequential Bayesian inference. Given a set of observed images $Y_t = [y_1, y_2, ..., y_t]$ at the t-th frame, the hidden state variable x_t

$$p(x_t|Y_{1:t}) \propto p(y_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|Y_{t-1})dx_{t-1} \tag{1}$$

where $p(x_t|x_{t-1})$ is the dynamic model between two sequential states, and $p(y_t|x_t)$ denotes observation model that estimates the likelihood of observing y_t at state x_t. The optimal state of all observing targets is obtained by the maximum a posteriori estimation over N samples at time t by

$$\widehat{x_t} = \arg\max_{x_t^i} p(y_t^i|x_t^i)p(x_t^i|x_{t-1}), i = 1, 2, ... \tag{2}$$

where x_t^i denotes the i-th sample of the state x_t, and y_t^i indicates the image patch estimated by x_t^i.

Dynamic model The affine warp model is used to model the target motion between two sequential frames. There are six parameter of the affine transformation used to model $p(x_t|x_{t-1})$. Let $x_t = [x_1, x_2, x_3, x_4, x_5, x_6]$, which denote shift in x, y translation, rotation, scale, and shear. Usually, the each dimension of $p(x_t|x_{t-1})$ is modeled by an independent Gaussian distribution(i.e. $p(x_t|x_{t-1}) = N(x_t; x_{t-1}, \Psi)$, where Ψ is a diagonal covariance matrix).

Observation model based on linear representation The global appearance of an object under different conditions (e.g. illumination and viewpoint change) is considered to lie approximately in a low dimensional space. Assume the target variable y is given by a deterministic function $f(T, a)$ with additive Gaussian noise as the following equation:

$$y = Ta + \epsilon \tag{3}$$

where tracking result $y \in R^d$ approximately lies in the linear span of T and is the zero mean, we denote the templates set as $T = [t_1, ..., t_n] \in R^{d \times n} (d >> n)$, containing n target templates such that each template $t_i \in R^d$ and $a = [a_1, a_2, ..., a_n]^T \in R^n$ is called a target coefficient vector. ϵ is the zero mean Gaussian random noisy term. Therefore, the observeration model is able to be extend to the following formulation:

$$p(y_t^i|x_t^i) = N(\epsilon; 0, I) \tag{4}$$

where I is a diagonal covariance matrix and ϵ is the residual of linear representation. Due to the $p(x_t|x_{t-1})$ drawing N particles from the the previous the particle of target and then resetting their weights to $1/n$, thus the Eq. 2 become the ordinary least square problem $||y - Ta||_2^2$.

2.2 Incremental subspace learning

As a classical method, the incremental tracking method [1] uses online PCA to update templates which can efficiently handle the problem that appearance change caused by in-plane rotation, scale, illumination variation and pose change. However, because of the intrinsic character of the representation based on PCA subspace, the IVT method is sensitive to partial occlusion, especially large occlusion. In PCA scheme, the underlying assumption is that the noisy term ϵ in Eq. 3 is Gaussian distributed with small variance. It is still a ordinary least square problem. Because of the orthogonality of bases T, the representation a can be estimated by $a = T^T Y$. And the noisy term $\|\epsilon\|_2^2 = \left\|Y - TT^T Y\right\|_2^2$.

Commonly, the noisy energy is very small because the PCA will hold the most variance after transforming input into a subspace. However, when the input is partly occluded by other objects, the IVT method would be failed in different conditions with partial occlusions which are non-Gaussian. Additionally, the IVT directly uses a new observation without any intervention for corruptions. As a result, it makes the observation model degraded in the situation with partial situation.

2.3 Sparsity regularization based tracker

Sparse representation has recently been extensively studied and applied in pattern recognition and computer vision, one of most successful applications is the sparse representation classification (SRC) in the face recognition problem [2]. In spite of existing different situations with various occlusion patterns, it always works well. Inspired by the SRC, Mei et al. [3] propose an algorithm by defining the tracking problem as finding the patch with minimum reconstruction error by sparse representation and handling occlusion with trivial templates by:

$$\min_{c} \frac{1}{2} \|y - Bc\|_2^2 + \lambda \|c\|_1 \ \ s.t.\ c \geq 0,\ B = [T\ I\ \ -I],\ c = [a\ e] \tag{5}$$

where y denotes an observer sample, T represents a sub dictionary of target templates, I indicates identity matrix as trivial template for error representation, a indicates the corresponding coefficient to target templates and the e is the coefficients of trivial templates. Commonly, only a few templates are necessary, whereas a lot of trivial templates are also needed. Therefore, the implementation make the dictionary too big to efficiently solve.

The Eq. 5 is able to obtain robust coefficients from the corrupted observation. In order to build a more robust model for object tracking, [7] and [8] propose an efficient and effective method to estimate different components from residual ϵ. They assume that the ϵ is a combination of Gaussian and Laplacian Distribution. Thus, they use huber loss function to replace least square function in the sparse representation or the ordinary least square problem.

$$f(x) = \begin{cases} x^2/2, & |x| \leq \lambda \\ \lambda|x| - \lambda^2/2, & (otherwise) \end{cases} \tag{6}$$

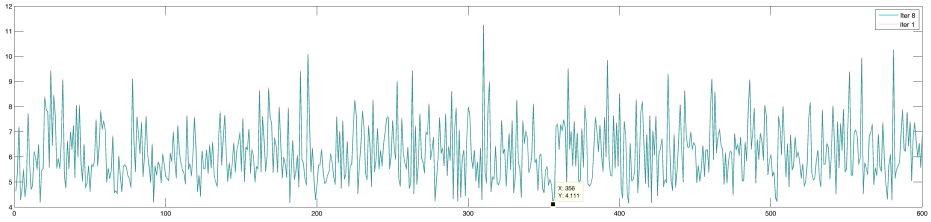
3 The proposed method

Many aforementioned methods employ the detection of pixel level outliers to solve the problem of degrading models. However, it is not an efficient and effective way to solve the problem of updating model. In this paper, we seek a different scheme which exploits a method without any prior about pixel level outliers, and update the observation model based on corrupted observations. Thus we only need a simple model which is good enough to estimate likelihood of particles.

3.1 Motivation

The Huber loss function is an efficient method to estimate different components from residual ϵ for building a more robust model. It assume that the ϵ is a combination of Gaussian and Laplacian distribution and decompose ϵ into $\mathbf{e} + \mathbf{s}$ iteratively. The relationship can be formulated as the following extend version of Eq.3:

$$y = Ta + \epsilon, s.t.\ \epsilon = \mathbf{e} + \mathbf{s} \tag{7}$$



(a) Distance Comparison

(b) Input and Outlier Detection

Fig. 1. Comparsion between Results with Different Iteration Numbers

where $\mathbf{e} = [e_1, ..., e_n]$ is Gaussian noisy and $\mathbf{s} = [s_1, ..., s_n]$ is sparse noisy followed Laplacian distribution. Thus, the ordinary least square problem using huber loss function [8] is equivalent to :

$$E = \min_{a,s} ||y - Ta - \mathbf{s}||_2^2 + \lambda \|\mathbf{s}\|_1 \tag{8}$$

To estimate the $\mathbf{e}$ and $\mathbf{s}$, the Eq. 8 needs to repeat two steps: detecting outlier pixels and then solving ordinary least square function without the influence of outlier pixels. Commonly, the function needs to repeat these two steps many times in order to obtain convergence. This method is capable of detecting pixel level outliers but really expensive in time.

However, in the repeated processes, the cost function (Eq. 8) update is usually very small. Fig. 1(a) shows a comparison between the cost value E after Iteration 1 and Iteration 8 for 600 particles. It is difficult to distinguish the difference between both lines. It is obvious to find that the tracking target is the 356-th particle because it has the lowest E. The 356-th particle and the outlier maps with different iteration numbers are shown in Fig. 1(b). It is obvious that the two outlier maps have very similar pattern, where the only difference is small.

Based on this fact, we argue that it is not necessary to use such a sophisticated method like the Huber loss function to exactly calculate which noisy of pixels are Gaussian or Laplacian distribution. For the sake of saving computation, we do not estimate these two different components accurately. According to the prior knowledge, the tiny ϵ is typically Gaussian distribution while the large ϵ is Laplacian distribution. Therefore, we can further simplify an assump-

tion that ***the ϵ is Gaussian distribution below a threshold, otherwise is Laplacian distribution***.

3.2 Observation Model

An image observation y_t^i can be represented by a subspace of the target object spanned by T if no occlusion occurs. Thus, many approaches use the reconstructed residual of each observed image patch to measure the observation likelihood by minimizing $||y^i - Ta^i||_2^2$:

$$p(\mathbf{y^i}|\mathbf{x^i}) = \exp(-\left\|\mathbf{y^i} - \mathbf{Ta^i}\right\|_2^2) \tag{9}$$

where we assume the y^i is centring. However, the Eq. 9 do not consider the impact introduced by complex noisy. It is necessary to account for partial occlusion in an appearance model. In [8], authors assume that a centered image observation can be represented by a linear combination of the PCA basis vectors and trivial templates. If partial occlusion occurs, the most likely image patch can be represented as a linear combination of PCA basis vectors and very few number of trivial templates (as illustrated by Fig. 1(b)). Thus, the precise localization of the tracked target can be benefited by penalizing the sparsity of trivial coefficients. But the problem of sparsity penalty is very computational expensive (i.e. Eq. 8).

We will propose the method to discriminate the type of noisy term using a threshold. As shown in Fig. 1(b), this scheme may be not perfect for pixel-wise occlusion, but it is good enough for the appearance model like Fig. 8. Specifically, we define a mask indicator M and a penalty vector W to point out Gaussian and Laplacian parts roughly:

$$m_i = \begin{cases} 1, |\epsilon_i| \leq \lambda \\ 0, (otherwise) \end{cases} \tag{10}$$

where m_i is a element of $M = [m_1, m_2, ..., m_n]$, which is a vector that indicates Gaussian elements of e. The λ is a threshold constant in all experiments of this paper.

$$w_i = \begin{cases} |\epsilon_i| - \lambda, |\epsilon_i| > \lambda \\ 0, \qquad (otherwise) \end{cases} \tag{11}$$

where w_i is a element of $W = [w_1, w_2, ..., w_n]$, which is a vector that indicates Laplacian elements of s. Eq. 11 is related to the soft-threshold function using subgradient (i.e. $sgn(x)(abs(x) - \lambda)$) for solving L_1 regularization problem [9].

$$p(\mathbf{y^i}|\mathbf{x^i}) = \exp(-\left\|M^i \odot (Y^i - Ta^i)\right\|_2^2 - \beta \left\|W^i\right\|_1) \tag{12}$$

where $\odot$ is the Hadamard product (element-wise product), and β is a penalty term. The first term accounts for reconstruction errors of the unoccluded proportion of the target object, and the second term indicates the impact of occluded part of the target object.

3.3 Update of Observation Model

Due to the least square loss function used, PCA is very sensitive to corrupted and contaminated observations. Even a few such outliers enable the quality of PCA output to be degraded. Unfortunately, for the tracking problem, outlier observations are frequent. In [10], researchers propose an online PCA method for outliers. The mechanism of probabilistic admission and rejection for new samples endows this method with the ability to be robust to the outliers. Technically, the method, in wild conditions, can be resistant to 50% breakdown point. However, this method cannot be employed in tracking problems directly. There are mainly two problems: ignoring the estimation of the mean vector and storing a full covariance matrix. The former is necessary for updating the appearance model. The latter one makes the method has to store a 1024×1024 covariance matrix and perform Eigen-decomposition on it while we have 32×32 patches as observations. It is not a small storage and computational burden for real time trackers.

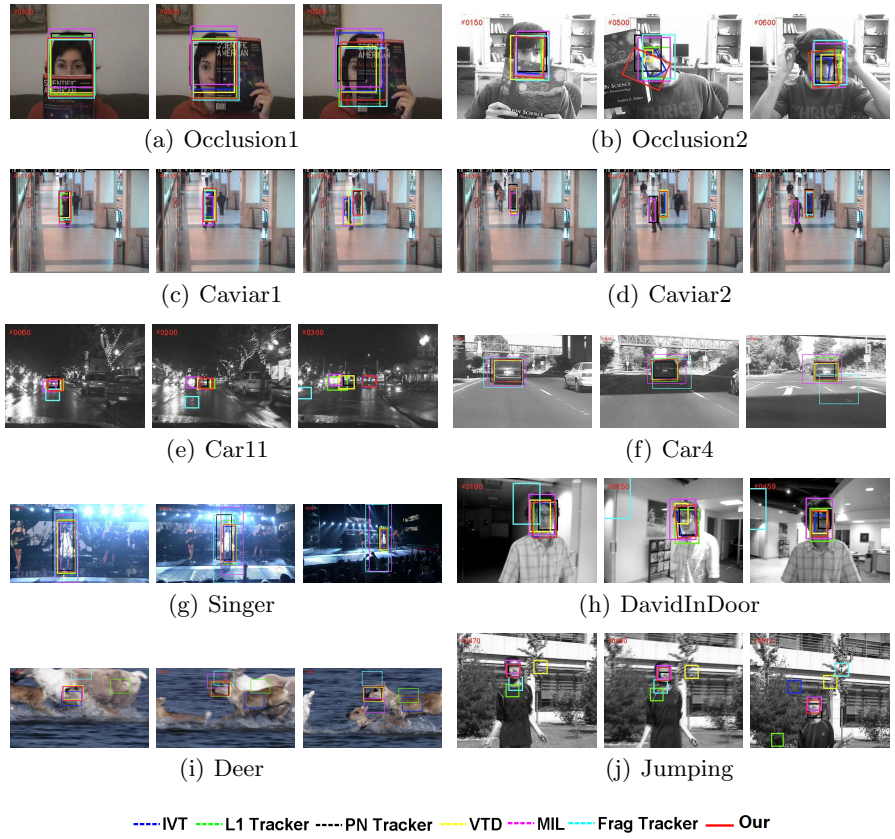
By combining the conventional incremental subspace learning method and the mechanism of probabilistic admission and rejection, we propose a robust incremental subspace learning in this paper. Nevertheless, we do not select the normalization of the data point by L_2-norm to estimate admissible probability and to update eigenvectors and eigenvalues, as same as [10]. We define the $\frac{\Sigma_{j=1}^{D} \hat{y}_{ij}^2}{\Sigma_{j=1}^{K} |U_j \hat{y}_i|^2}$ as a measure to estimate acceptable probability. Once a new sample is accepted, it would be process by conventional incremental subspace learning as the same in [1]. Such a method has a theoretical performance guarantee under the noisy case. For instance, in [10] the strategy works well even in the situation of 30% outlier fraction in the experiment, while the online PCA fails in the situation of 5% outlier fraction.

Algorithm 1 Robust Incremental Subspace Learning

Input: Data Sequence $[y_1, ..., y_b] \in R^{D \times b}$,buffer size b, eigenvectors $U \in R^{D \times K}$, eigenvalues $\Sigma \in R^{D \times D}$ and the mean vector $I \in R^D$.
Output: updated eigenvectors U', eigenvalues Σ' and the mean vector I'.
Initialization: 1) y' = []

repeat
a) Centering the data point as $\hat{y}_i = y_i - I$;
b) Estimating the energy of data point as $e_i = \Sigma_{j=1}^{D} \hat{y}_{ij}^2$
c) Calulate the variance of y_i along the direction eigenvectors U: $\theta_i = \Sigma_{j=1}^{K} |U_j \hat{y}_i|^2$
d) Accept y_i with probability θ_i / e_i
e) If y_i is accepted, $y' = [y'\ \hat{y}_i]$

until $i > b$
2) Perform incremental subspace learning [1] on y' and obtain new eigenvectors U', eigenvalues Σ' and the mean vector I'.



(a) Occlusion1 (b) Occlusion2

(c) Caviar1 (d) Caviar2

(e) Car11 (f) Car4

(g) Singer (h) DavidInDoor

(i) Deer (j) Jumping

(k) Legend

Fig. 2. Sample tracking results on ten challenging image sequences. This figure demonstrates the results of the IVT, L1 tracker, PN Tracker, VTD, MIL, Frag Tracker and the proposed method.

4 Experiments

Speed The proposed method is implemented in both MATLAB and C. The MATLAB versions runs at 10 fps on an Air-Mac with Core i5 1.86GHZ CPU and 4GB memory, and the C version runs at 25-30 fps on Xeon E5305 CPU and 16GB memory (Tab .1). Please note that, our C version is single thread, and other comparison methods are usually on multi-thread (Matlab's default matrix operation option).

Dataset We evaluate the proposed tracker against ten state-of-the-art algorithms qualitatively and quantitatively, using the source codes provided by the authors for fair comparisons, including the IVT [1], FragTrack(FT) [11], MILTrack [12], VTD [13], PN [14], TLD [15], APGL1 [6], ASLSA [16], MTT [17],

Table 1. Comparison of computational costs

Method	L1APG	ONNDL	OSPT	ASLS	Ours (C Version)
FPS	14.5	1.4	4.7	1.3	25

ONNDL[7], OSPT [8], SCM [18], and LSAT [19]. In the evaluation, we use fifteen challenging image sequences from prior works [1, 3, 12, 13, 20] and CAVIAR dataset. The challenging factors of these sequences include partial occlusion, background clutter, motion blur, illumination and pose variation.

Experimental setting For each sequence, the initial location of the target object is manually labelled. For the sake of representation based on robust incremental subspace learning, each image observation is normalized to 32×32 pixels patch and 16 eigenvectors are selected in all experiments. As the trade-off between computational efficiency and effectiveness, 600 particles are used and the incremental subspace learning updates parameters every 5 frames. The noisy term threshold λ is set to 0.1 to all experiments.

4.1 Qualitative Evaluation

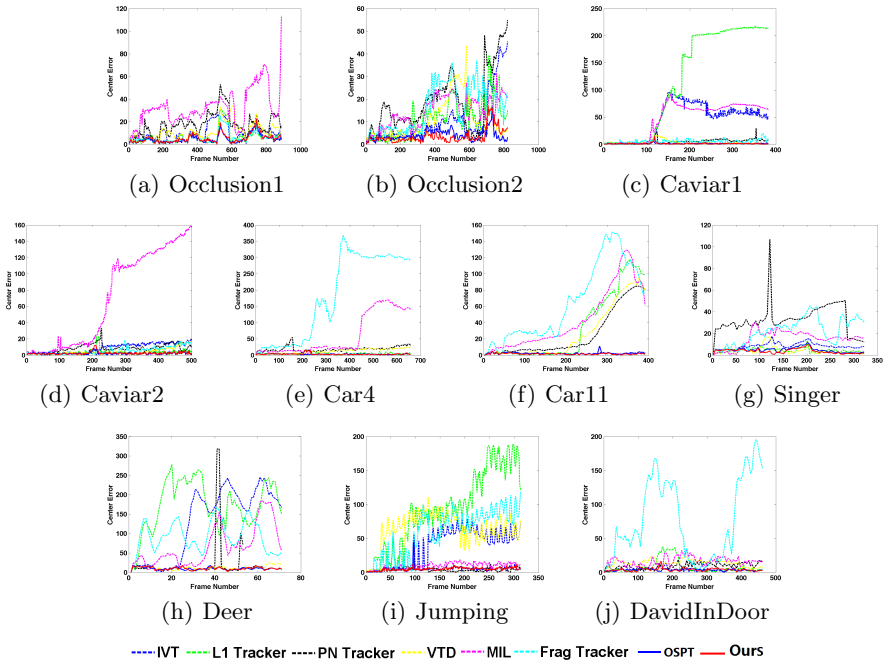
All the sequence images in our experiment are simulating situations of three categories: heavy occlusion, illumination change and fast motion. With limited space available, we only give a qualitative comparison as shown in Fig. 2 with some key frames of ten sequences.

For the situation under heavy occlusions, such as *Occlusion*1 sequence that the face is occluded by a magazine significantly [11], our approach, FragTrack and L1 methods perform better as shown in Fig. 2(a) due to taking partial occlusion into account. Before the magazine covers the face, all trackers do a fine job. However, after that many of them only track the face inaccurately. In *Occlusion*2 sequence that simulates a more complex situation by appearing occlusion and in-plane rotation at the same time. As shown in Fig. 2(b), although all tracker work well in the partial occlusion at about frame 150. They fail in the situation combining in-plane rotation and partial occlusion at about frame 500. *Carviar*1 and *Caviar*2 are surveillance videos which are challenging as they contain scale change, partial occlusion and similar target. The L1 and IVT trackers drift away from the target at frame 225 in Fig. 2(d) or fail totally at frame 125 in Fig. 2(c) after it is occluded by a similar object. Our method is succeeded in solving the challenges.

For the situation under illumination change, such as *Car*4 in which the illumination changes abruptly due to the shallow of entrance and exit of a tunnel. All tracker work well before the car enters the tunnel at about frame 160 in the Fig. 2(f). However, after that only our method and IVT can track the car accurately and others track the target with drift or miss the target totally. In *Car*11 sequences in where the road environment is very dark with background light, we notice the IVT approach and the proposed method perform better than

the other methods whereas the other methods drift away when the abrupt illumination change occurs (e.g frame 200 in Fig. 2(e)) or when the similar objects are close to the target (e.g frame 300 in Fig. 2(e)). In *DavidIndoor* sequence, recorded in an indoor environment, we need to track a moving face with illumination, scale and out-plane rotation changes. Most methods drift from frame 160 in Fig. 2(h) because of out-plane rotation . But some of them can recover this after several frames. In *Singer* sequence, complicated illumination changes make the tracking task more difficult. Our tracker is able to track the target more precisely than others as shown in Fig. 2(g).

For the fast motion situation, such as *Deer* sequence in which the target is a running deer with rapid changes in appearance, our method and VTD method work better than other methods. In *Jumping* sequence, the trackers have to face the challenge of appearance change caused by motion blur. The MIL, PN tracker and the proposed method can track the target even the target become blurred.



(a) Occlusion1 (b) Occlusion2 (c) Caviar1

(d) Caviar2 (e) Car4 (f) Car11 (g) Singer

(h) Deer (i) Jumping (j) DavidInDoor

(k) Legend

Fig. 3. Quantitative Comparison between the Average Center Error among Different Methods in 10 Datasets

Table 2. Average center location error. The best three results are shown in red, blue and green fonts seperatively.

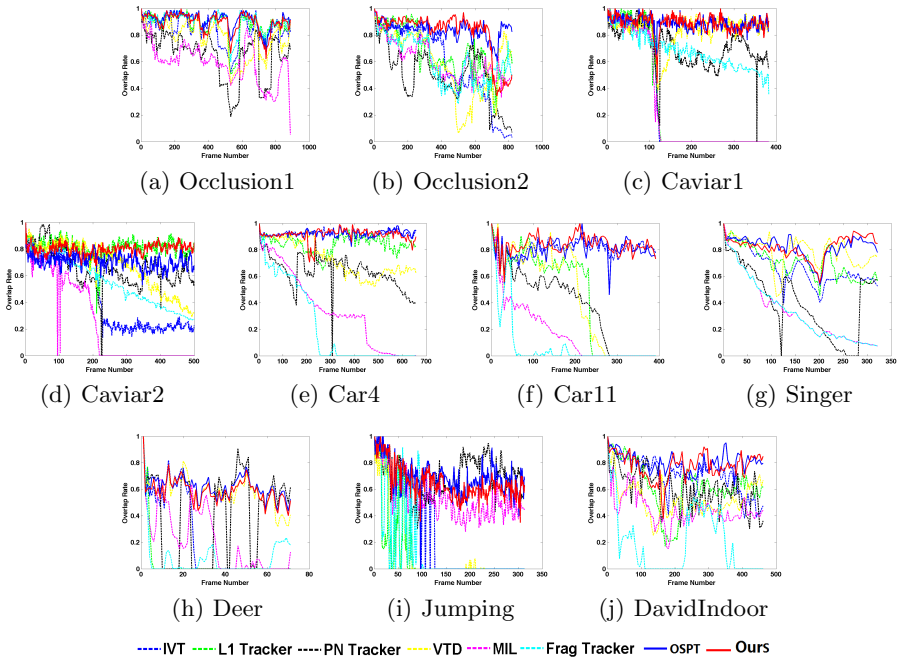
Sequence	FT	IVT	ONNDL	VTD	TLD	APGL1	MTT	LSAT	SCM	ASLAS	OSPT	Ours
Occlusion1	5.6	9.2	5.0	11.1	17.6	6.8	14.1	5.3	3.2	10.8	4.7	5.3
Occlusion2	15.5	10.2	8.6	10.4	18.6	6.3	9.2	58.6	4.8	3.7	4	3.5
Caviar1	5.7	45.2	3.2	3.9	5.6	50.1	20.9	1.8	0.9	1.4	1.7	1.4
Caviar2	5.6	8.6	4.4	4.7	8.5	63.1	65.4	45.6	2.5	62.3	2.2	2.4
Caviar3	116.1	66	63.7	58.2	44.4	68.6	67.5	55.3	2.2	2.2	45.7	3.2
DavidOut	90.5	53	53.3	61.9	173	233.4	65.5	101.7	64.1	87.5	5.8	7.9
DavidIn	148.7	3.1	6.0	49.4	13.4	10.8	13.4	6.3	3.4	3.5	3.2	3.9
Singer1	22	8.5	9.3	4.1	32.7	3.1	41.2	14.5	3.7	5.3	4.7	3.5
Car4	179.8	2.9	6.0	12.3	18.8	16.4	37.2	3.3	3.5	4.3	3	3.2
Car11	63.9	2.1	1.4	27.1	25.1	1.7	1.8	4.1	1.8	2	2.2	1.5
Deer	92.1	127.5	8.3	11.9	25.7	38.4	9.2	69.8	36.8	8	8.5	10
Football	16.7	18.2	19.6	4.1	11.8	12.4	6.5	14.1	10.4	18	33.7	7
Jumping	58.4	36.8	79.1	63	3.6	8.8	19.2	55.2	3.9	39.1	5	4.8
Owl	148	141.4	27.8	86.8	8.2	104.2	184.3	110.7	7.3	7.6	47.4	6.2
Face	48.8	69.7	29.5	141.4	22.3	148.9	127.2	16.5	125.1	95.1	24.1	12.3
Average	67.8	40.2	21.7	36.7	28.6	51.5	45.5	37.5	18.2	23.4	13.1	5

4.2 Quantitative Evaluation

There are two different evaluation for our quantitative evaluation: the difference between the predicated and the ground truth center locations, and the overlap rate with ground truth. The results are summarized in Tab. 2 and Tab. 3, respectively. The Tab. 2 reports the average center location errors in pixels, where the value smaller the result more accurate. Given the tracking result of each frame R_t and the corresponding ground truth R_G, the overlap rate is defined as $\frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$. Tab. 3 reports the average overlap rates, where the value larger the result more accurate. For each video sequence (i.e., each row), we show the best result in red, second best in blue, third best in green. We also report the central-pixel errors and the overlap rates frame-by-frame for each video sequence in Fig. 3 and Fig.4 respectively. In terms of the overlap rate, our method is always among the best three in 13 of the 15 sequences. With respect to the central-pixel error, our method is among the best two in 13 of the 15 sequences. For the other two sequences, the gaps are quite small. We believe they can be negligible in practical applications. Looking at the overall results. Our algorithm achieves the lowest tracking errors in the most of all sequences, and archives the highest overlap rate.

5 Conclusion

In this paper, we propose a robust incremental visual tracking method which take probabilistic admission into account for observation model updating and take partial occlusion into account for object tracking. Either robust incremental



(a) Occlusion1 (b) Occlusion2 (c) Caviar1

(d) Caviar2 (e) Car4 (f) Car11 (g) Singer

(h) Deer (i) Jumping (j) DavidIndoor

(k) Legend

Fig. 4. Quantitative Comparison between the Overlap Rates among Different Methods in 10 Datasets

Table 3. Average overlap rate. The best three results are shown in red, blue and green fonts seperatively.

Sequence	FT	IVT	ONNDL	VTD	TLD	APGL1	MTT	LSAT	SCM	ASLAS	OSPT	Ours
Occlusion1	0.90	0.85	0.89	0.77	0.65	0.87	0.79	0.90	0.93	0.83	0.91	0.89
Occlusion2	0.60	0.59	0.54	0.59	0.49	0.70	0.72	0.33	0.82	0.81	0.84	0.82
Caviar1	0.68	0.28	0.67	0.83	0.70	0.28	0.45	0.85	0.91	0.90	0.89	0.89
Caviar2	0.56	0.45	0.46	0.67	0.66	0.32	0.33	0.28	0.81	0.35	0.71	0.80
Caviar3	0.13	0.14	0.13	0.15	0.16	0.13	0.14	0.58	0.87	0.82	0.25	0.85
DavidOut	0.39	0.52	0.71	0.42	0.16	0.05	0.42	0.36	0.46	0.45	0.77	0.74
DavidIn	0.09	0.69	0.56	0.23	0.5	0.63	0.53	0.72	0.75	0.77	0.76	0.78
Singer1	0.34	0.66	0.58	0.79	0.41	0.83	0.32	0.52	0.85	0.78	0.82	0.82
Car4	0.22	0.92	0.88	0.73	0.64	0.7	0.53	0.91	0.89	0.89	0.92	0.91
Car11	0.09	0.81	0.82	0.43	0.38	0.83	0.58	0.49	0.79	0.81	0.81	0.84
Deer	0.08	0.22	0.60	0.58	0.41	0.45	0.6	0.35	0.46	0.62	0.61	0.59
Football	0.57	0.55	0.44	0.81	0.56	0.68	0.71	0.63	0.69	0.57	0.62	0.69
Jumping	0.14	0.28	0.06	0.08	0.69	0.59	0.3	0.09	0.73	0.24	0.69	0.65
Owl	0.09	0.22	0.46	0.12	0.6	0.16	0.09	0.13	0.79	0.78	0.48	0.79
Face	0.39	0.44	0.56	0.24	0.62	0.14	0.26	0.69	0.36	0.21	0.68	0.76
Average	0.35	0.51	0.56	0.5	0.51	0.49	0.45	0.52	0.74	0.66	0.72	0.79

subspace method or outliers detection for observations didn't introduce new computational burden. Thus they are positive to make the method be real time. Both quantitative and qualitative evaluations on challenging image sequences demonstrate that the proposed tracking method outperforms several state of the art methods. In the future work, the λ in Eq. 10 is supposed to be measured dynamically according to the real scenes.

Acknowledgement. The authors would like to thank the anonymous reviewers for constructive comments that helped in improving the quality of this manuscript and Dr. NaiYan Wang for useful discussions.

References

1. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77** (2008) 125–141
2. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31** (2009) 210–227
3. Mei, X., Ling, H.: Robust visual tracking using 1 minimization. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 1436–1443
4. Kim, S.J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale l 1-regularized least squares. Selected Topics in Signal Processing, IEEE Journal of **1** (2007) 606–617
5. Mei, X., Ling, H., Wu, Y., Blasch, E., Bai, L.: Minimum error bounded efficient 1 tracker with occlusion detection. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1257–1264
6. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1830–1837
7. Wang, N., Wang, J., Yeung, D.Y.: Online robust non-negative dictionary learning for visual tracking. In: Computer Vision (ICCV), 2013 IEEE International Conference on. (2013) 657–664
8. Wang, D., Lu, H., Yang, M.H.: Online object tracking with sparse prototypes. Image Processing, IEEE Transactions on **22** (2013) 314–325
9. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Foundations and Trends® in Machine Learning **4** (2012) 1–106
10. Feng, J., Xu, H., Mannor, S., Yan, S.: Online pca for contaminated data. In: Advances in Neural Information Processing Systems. (2013) 764–772
11. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 798–805
12. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 983–990
13. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 1269–1276
14. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 49–56

15. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 1409–1422
16. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1822–1829
17. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 2042–2049
18. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1838–1845
19. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 1313–1320
20. Wu, Y., Ling, H., Yu, J., Li, F., Mei, X., Cheng, E.: Blurred target tracking by blur-driven tracker. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1100–1107