Online Multiple Instance Joint Model for Visual Tracking

Longyin Wen[1] Zhaowei Cai[1] Menglong Yang[1] Zhen Lei[1,2] Dong Yi[1,2] Stan Z. Li[1,2]*
[1]CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences
[2]China Research and Development Center for Internet of Thing
{lywen,zwcai,mlyang,zlei,dyi,szli}@cbsr.ia.ac.cn

Abstract

Although numerous online learning strategies have been proposed to handle the appearance variation in visual tracking, the existing methods just perform well in certain cases since they lack effective appearance learning mechanism. In this paper, a joint model tracker (JMT) is presented, which consists of a generative model based on Multiple Subspaces and a discriminative model based on improved Multiple Instance Boosting (MIBoosting). The generative model utilizes a series of local constructed subspaces to update the Multiple Subspaces model and considers the energy dissipation of dimension reduction in updating step. The discriminative model adopts the Gaussian Mixture Model (GMM) to estimate the posterior probability of the likelihood function. These two parts supervise each other to update in multiple instance way which helps our tracker recover from drift. Extensive experiments on various databases validate the effectiveness of our proposed method over other state-of-the-art trackers.

1. Introduction

Visual tracking is one of the most fundamental and important problems in computer vision, and it has widely practical applications, such as human-computer interaction, video surveillance, robotics and etc. The tracker is required to capture the position and size of the target, but unfortunately, it is always influenced by many annoying factors, including occlusion, illumination variation and pose changes.

Generative model [10, 16, 1, 11] and discriminative model [2, 3, 6, 18] are two major methods to address the above challenges. The generative model formulates the tracking task as a regression problem which searches the point with maximum likelihood. The discriminative model formulates the tracking task as a classification problem which focuses on the difference between the target and the background. Lim et al. proposed a generative model [10] to acquire the features of the target by means of incremental PCA, which performs well in rigid or limited deformable motion because it constructs an adaptive appearance model of the target online. H. Grabner et al. proposed a discriminative model based on the online boosting [6], which has good performance in tracking because of its strong discriminative ability to recognize the target from the background. However, once drift problem happens, these generative and discriminative models cannot recover since they directly view their outputs as the positive sample for updating and the accumulated errors will finally result in their failure. To handle the drift problem, B.Babenko et al. proposed a robust discriminative tracker based on multiple instance learning [3]. The multiple instance mechanism ensures the tracker to extract the 'true' samples for updating to alleviate drifting problem even when occlusion occurs.

Although the above methods perform well in some scenarios, it is relatively easy to lose the target because of the lack of mutual supervision system. Some researchers realized this problem and have developed several trackers with combined models. The trackers proposed in [21, 12, 13] combine several models with different views for tracking and adopt the co-training framework for updating. They achieve more stable performances than the single generative or discriminative tracker as the result of mutual supervision. In [17, 8], three models are incorporated in the tracker, which makes the tracker robust to certain situation. However, these trackers with combined models also suffer from drift problem and their updating samples are collected at the maximum posterior probability point with hard label (*i.e.* $y = \{-1, +1\}$), even when the confidences are low [7].

To deal with the problems existing in current trackers, we develop a novel joint model to accomplish the tracking task which is constituted by the Incremental Multiple Instance Multiple Subspaces Learning (*IMIMSL*) and the improved Multiple Instance Boosting (MIBoosting). The two parts supervise each other and they are updated in the multiple instance way. As discussed above, the mutual supervision system will provide our tracker more stability and the multiple instance updating strategy will help our tracker recover

[*] Stan Z. Li is the corresponding author

978-0-7695-4797-8/12 $26.00 © 2012 IEEE
DOI 10.1109/AVSS.2012.52

IEEE computer society

from drift. Meanwhile, we assign the unlabeled data for updating with weights qualifying the probabilities of samples instead of the hard label, which will successfully avoid the manual threshold selection in hard label assignment and resist the noise more effectively.

For the generative model that focuses on the target feature space, our proposed *IMIMSL* utilizes the subspace for updating, and the subspace is constructed by the representative weighted samples. To better acquire the target features, the energy dissipation of dimension reduction is taken into account in the updating step, which is always ignored in the previous proposed methods [10, 21]. For the discriminative model that concentrates on the difference between target and background, the improved MIBoosting is proposed. Different from the method in [3], we apply GMM to estimate the likelihood function $p(f(x)|y = \pm 1)$, which can represent the practical probability of the likelihood function more precisely than the single Gaussian model.

2. Sequential Inference with Multiple Instance Supervision

The tracking task is formulated as a state estimation problem and the motion process is assumed to be a Markovian state transition process. Let $Z_i = \{z_1, \cdots, z_i\}$ represent the observation data up to time i. x_i is the state vector at time i, which contains the position and size of the target. In our tracker, the state vector is composed by *X*-axis, *Y*-axis, target width, target height. The posterior probability is estimated as the recursive equation:

$$p(x_t|Z_t) \propto p(z_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|Z_{t-1}) dx_{t-1} \quad (1)$$

where $p(z_t|x_t)$ is the likelihood of the candidate samples. $p(x_t|x_{t-1})$ is the state transition probability and $p(x_{t-1}|Z_{t-1})$ is the state estimation probability given all observations. Similar as [3], we adopt the simplest greedy Maximum a Posteriori Probability (**MAP**) strategy to solve the above equation, where the motion model is specified as $p(x_t|x_{t-1}) = 1$, if $\|l_t - l_{t-1}\| < r$, otherwise $p(x_t|x_{t-1}) = 0$, where l_t is the position of the target at time t, r is the search radius and the distance is the Euclidean distance.

The appearance probability of the state x_t is constituted by two parts, that is $p(z_t|x_t) = p(z_t^g|x_t)p(z_t^d|x_t)$. The term $p(z_t^g|x_t)$ is the probability produced by *IMIMSL* model and $p(z_t^d|x_t)$ is given out by improved MIBoosting model. The two parts of the joint model select multiple updating samples for each other and the selected samples are packaged into bags for updating. Meanwhile, the noise is controlled efficiently by introducing the weights of updating samples instead of the hard positive or negative label, which are generated by the view's counterpart. In our work, *IMIMSL* model utilizes the gray value of images as the feature and the improved MIBoosting model extracts haar-like features [4]. In the following sections, the two parts of the joint model will be poured out in details.

3. Generative Model with Multiple Subspaces

In the proposed multiple subspaces learning method, the packaged positive bags collected from the neighborhood of the tracker output are utilized for updating. The maximum probability instance within the bag is utilized to represent the bag for updating. The positive bag B is consisted by the instance set $\{I_1, \cdots, I_n\}$ and $p(I_i)$ is the probability of the i^{th} instance within the bag to be positive, which is produced by the discriminative model. The weight of the maximum probability instance is calculated as: $w_{I^*} = p(B)p(I^*)$, where I^* is the maximum likelihood instance and the Noisy-OR model is utilized to estimate the probability of the bag, that is $p(B) = 1 - \prod_{i=1}^{n}(1 - p(I_i))$. Finally, the updating instance is selected in this way with the corresponding assigned weight.

3.1. Multiple Linear Subspaces Model

The appearance manifold of the target is highly nonlinear and it is hard for us to estimate it directly. However, multiple low dimension subspaces can be applied to approximate the manifold of the target. In this section, we propose a novel incremental learning strategy of the multiple local subspaces which utilizes the combined samples in adjacent frames rather than individual ones for updating. This learning strategy learns the features of the target more efficiently and reduces the homogeneous noise contained in the samples. Let $\mathcal{M} = \{\Omega_1, \cdots, \Omega_L\}$ represent the multiple subspaces of the target and Ω_i, $i \in \{1, \cdots, L\}$ indicate the local subspace, where L is the total number of multiple subspaces. An observed instance z is a d-dimension image vector. Let $\Omega_i = (\mu_i, V_i, \Lambda_i, W_i, n_i)$, where μ_i, V_i, Λ_i, W_i and n_i represent the mean vector, the eigenvectors, the eigenvalues, the set of sample weights and the number of samples respectively. The multiple subspaces learning strategy is detailed in Algorithm 1.

Approximate D representative instances are compressed into a local subspace. The subspace construction process can be completed by matrix *SVD* or the efficient *EM* algorithm proposed in [19]. A η-truncation is utilized to decide the reduction dimension of the subspace to maintain the energy, that is $q = \arg\min_i \left(\frac{\sum_i \lambda_i}{tr(\Lambda)} \geq \eta\right)$. To evaluate the probability of the sample, we utilize the maximum probability of the L_c nearest subspaces (L_c is set as 3 in the experiments). Set the observation variable of the *IMIMSL* model to be z_t^g and $y^t = (y_1^t, \cdots, y_q^t) = V_t^T(z_t^g - \mu_t)$, where t is the time index, V_t and μ_t are the eigenvectors and the eigenvalues of the subspace. The probability of the

Algorithm 1 Online Multiple Subspaces Learning Algorithm

```
Input: $(\gimel, \mathcal{M}, D, L)$
$\gimel = \{\mathcal{I}_1, \cdots, \mathcal{I}_i, \cdots\}$: a sequence of updating samples; $\mathcal{M} = \emptyset$: the multiple subspaces; $D$: the number of samples for the updating subspace; $L$: the total number of subspaces. Set $U$ to be the valid updating pool, symbol $|U|$ represents the number of samples in the pool and $k$ is the index of the updating sample.
Output: $\mathcal{M} = (\Omega_1, \cdots, \Omega_L)$: multi-local subspaces.
while $k \leq |\gimel|$ do
    if $|U| < D$ then
        Add $\mathcal{I}_k$ to the updating pool $U$.
    else
        if There exists an empty subspace of $\mathcal{M}$ then
            Construct the subspace $\Omega_n$ with the samples in pool $U$, Add $\Omega_n$ to $\mathcal{M}$ and clear the updating pool $U$.
        else
            Construct the subspace $\Omega_n$ by the samples in pool $U$, and calculate the similarity between subspaces, that is $\{p, q\}^* = \arg\max Sim(\Omega_p, \Omega_q)$, where $\Omega_p, \Omega_q \in \{\Omega_1, \cdots, \Omega_L\} \bigcup \{\Omega_n\}$, $p \neq q$. $\Omega_m = \Omega_p \bigcup \Omega_q$, which means the subspace merging process, and replace the subspaces $\Omega_p$ and $\Omega_q$ with $\Omega_m$. Clear the updating pool $U$.
        end if
    end if
    $k = k + 1$.
end while
```

Algorithm 2 The Subspace Updating Algorithm

```
Update the mean value of the subspaces, $\mu^{(k+l)} = \gamma\mu^{(k)} + (1-\gamma)\mu^{(l)}$, where $\gamma = \frac{\sum_{\omega_i \in W_k} \omega_i}{\sum_{\omega_i \in W_{k+l}} \omega_i}$.
Set $\rho = \frac{\sum_{\omega_i \in W_k} \omega_i^2}{\sum_{\omega_i \in W_{k+l}} \omega_i^2}$. Get the observation covariance matrix $S^{(k+l)} \approx (\rho\sigma_k^2 + (1-\rho)\sigma_l^2)I + LL^T$. For simplicity, decompose the matrix $L^T L$ instead of matrix $LL^T$.
Set $Q = L^T L = \begin{pmatrix} \Sigma & \beta \\ \beta^T & \alpha \end{pmatrix}$, the size of matrix $Q$ is $(q+1) \times (q+1)$, where $\Sigma = \begin{pmatrix} \Sigma_1 & A \\ A^T & \Sigma_2 \end{pmatrix}$. Decompose $Q$ as: $Q = U\mathit{\Gamma} U^T$, where $\mathit{\Gamma} = diag\{\xi_1, \xi_2, \cdots, \xi_{q+1}\}$, $U^T U = I$. Then $V_{q_k+q_l+1} = LU\mathit{\Gamma}^{-\frac{1}{2}}$, where matrix $V_{q_k+q_l+1} = [v_{1,k+l}, \cdots, v_{q_k+q_l+1,k+l}]$ is composed by the first $q_k + q_l + 1$ eigenvectors of the covariance matrix $S^{(k+l)}$.
The observation covariance matrix is represented as: $S^{(k+l)} = (\rho\sigma_k^2 + (1-\rho)\sigma_l^2) + \sum_{i=1}^{q_k+q_l+1} \xi_i v_{i,k+l} v_{i,k+l}^T$. The first $q_k + q_l + 1$ eigenvalues of the covariance matrix can be updated as $\lambda_{i,k+l} = {\sigma^{(k+l)}}^2 + \xi_i$, and the sigma value is updated as $\sigma_{k+l}^2 = \frac{1}{d-q_{k+l}}(\sum_{i=q_{k+l}+1}^{q_k+q_l+1} \lambda_{i,k+l} + (d - q_k - q_l - 1){\sigma^{(k+l)}}^2)$, where ${\sigma^{(k+l)}}^2 = \rho\sigma_k^2 + (1-\rho)\sigma_l^2$, and $q_{k+l} = \arg\min_i(\frac{\sum_i \lambda_{i,k+l}}{\sum_{j=1}^{q_k+q_l+1} \xi_j} \geq \eta)$.
```

candidate sample to be positive is expressed [14] as:

$$p(z_t^g|x_t) = \left[\frac{\exp\left(-\frac{\varepsilon^2(x_i)}{2\sigma^2}\right)}{(2\pi\sigma^2)^{(d-q)/2}}\right]\left[\frac{\exp\left(-\sum_{i=1}^{q} \frac{{y_i^t}^2}{2\lambda_i}\right)}{(2\pi)^{q/2}\Pi_{i=1}^{q}\sqrt{\lambda_i}}\right] \quad (2)$$

The symbol $\varepsilon(x_i)$ in the first item is the residual of the samples projected to the subspace, which is calculated as: $\varepsilon(x_i) = \|x_i - VV^T x_i\|$.

Subspace Similarity The similarity of the two subspaces is estimated as the weighted combination of the angle measure $Sim_\alpha(\Omega_1, \Omega_2)$ and the compactness measure $Sim_c(\Omega_1, \Omega_2)$, which is defined as: $Sim(\Omega_1, \Omega_2) = Sim_\alpha(\Omega_1, \Omega_2) + \omega Sim_c(\Omega_1, \Omega_2)$, while ω is the trade-off between the two similarity measures and we set $\omega = 0.18$ in our experiments. Please refer to [21] for more details.

Subspace Updating The core problem in subspace incremental learning is the updating strategy. Our proposed strategy utilizes the subspaces for updating, namely merging the two most similar subspaces into one subspace. Derived from the basic equation of the maximum likelihood solution of traditional probability principal component analysis and taking the weights of the samples into account, the mean value $\mu^{(k)}$ and covariance matrix $S^{(k)}$ of the subspace are represented as: $\mu^{(k)} = \frac{1}{\sum_{\omega_i \in W_k} \omega_i} \sum_{\omega_i \in W_k} \omega_i z_i$ and $S^{(k)} = \frac{1}{\sum_{\omega_i \in W_k} \omega_i^2} \sum_{\omega_i \in W_k} \omega_i^2 (z_i - \mu^{(k)})(z_i - \mu^{(k)})^T$, where z_i is the observations. We can get the covariance matrix of the merged subspace:

$$S^{(k+l)} \approx \frac{\sum_{\omega_i \in W_k} \omega_i^2}{\sum_{\omega_i \in W_{k+l}} \omega_i^2} S^{(k)} + \frac{\sum_{\omega_i \in W_l} \omega_i^2}{\sum_{\omega_i \in W_{k+l}} \omega_i^2} S^{(l)} + yy^T \quad (3)$$

where we set the relations: $y = (\frac{\sum_{\omega_i \in W_k} \omega_i^2}{\sum_{\omega_i \in W_{k+l}} \omega_i^2}(1-\gamma)^2 + \frac{\sum_{\omega_i \in W_l} \omega_i^2}{\sum_{\omega_i \in W_{k+l}} \omega_i^2}\gamma^2)^{\frac{1}{2}}(\mu^{(k)} - \mu^{(l)})$ and $\gamma = \frac{\sum_{\omega_i \in W_k} \omega_i}{\sum_{\omega_i \in W_{k+l}} \omega_i}$.

Furthermore, the covariance matrix can be decomposed as the following $S_k = \sigma_k^2 I + \sum_{i=1}^{q_k}(\lambda_{i,k} - \sigma_k^2) v_{i,k} v_{i,k}^T$ and we have the equation: $\sigma_k^2 = \frac{1}{d_k - q_k} \sum_{q_k+1}^{d_k} \lambda_{i,k}$. We reformulate the equation above to get:

$$S^{(k+l)} \approx (\rho\sigma_k^2 + (1-\rho)\sigma_l^2)I + LL^T \quad (4)$$

where $L = [\sqrt{\rho(\lambda_{1,k} - \sigma_k^2)} v_{1,k}, \cdots, \sqrt{\rho(\lambda_{q_k,k} - \sigma_k^2)} v_{q_k,k} \sqrt{\bar{\rho}(\lambda_{1,l} - \sigma_l^2)} v_{1,l}, \cdots, \sqrt{\bar{\rho}(\lambda_{q_l,l} - \sigma_l^2)} v_{q_l,l}, y]$ and $v_{i,k}$, $\lambda_{i,k}$, σ_k and $v_{i,l}$ $\lambda_{i,l}$, σ_l are the i^{th} eigenvector, i^{th} eigenvalue, energy dissipation in dimension reduction of the covariance matrix S_k and S_l respectively, and $\rho = \frac{\sum_{\omega_i \in W_k} \omega_i^2}{\sum_{\omega_i \in W_{k+l}} \omega_i^2}$, $\bar{\rho} = 1 - \rho$.

Considering the computation complexity, we decompose $L^T L$ instead of matrix LL^T. Let $Q = L^T L$. The size of matrix Q is $q \times q$, where $q = q_k + q_l + 1$. Then the subspace updating process can be done efficiently by decomposing the matrix Q and the process is detailed in Algorithm 2. Specifically, when the updating subspace is constructed by only one individual sample, the proposed updating process is also applicable. In that condition, the covariance matrix $S = 0$, the mean value is equal to the feature value of the sample, and the reduction dimension equals 1. Please refer the supplementary material for more details about the deducing process of the subspace updating.

3.2. Performance Evaluation

To verify the effectiveness of the model, we conduct a experiment in the sequences Sylvester and Minghsuan [10] with severe pose changes and challenging lighting respectively. Totally 4 subspaces are adopted in our model and we use the smallest projection error of 3 center nearest subspaces to be the measure. Every 3 frames are combined

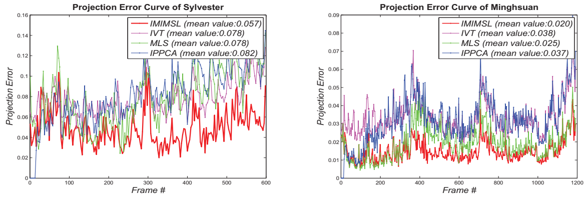

Figure 1. The projection error curve of the our *IMIMSL* model and three other methods in two sequences.

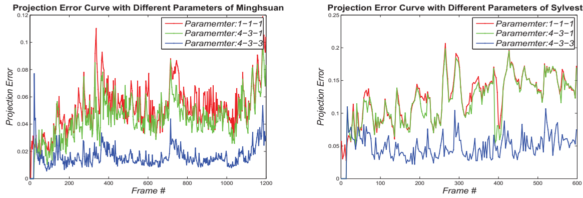

Figure 2. The projection error curve of our proposed method with different parameters of the test sequences. The symbol a-b-c represents the model has totally a subspaces and calculates the maximum probability of b-nearest subspaces, and every constructed subspace contains c combined samples.

to construct a new subspace for updating. The comparison is conducted with three other state-of-the-art learning strategies: IVT [10], the Multiple Linear Subspaces (MLS) model [21] and IPPCA [15]. The parameters of our model are fixed in these two sequences. The parameters of IVT, MLS and IPPCA are set as the default ones in their papers or codes. The comparison of projection error is shown in Figure 1. In both sequence, IVT and IPPCA have the worst performances, because they only construct a subspace which is updated with single sample. With the introduction of the multiple linear subspaces which are updated with combined samples, MLS outperforms IVT and IPPCA. However, its results are not as accurate as ours since our *IMIMSL* considers the energy dissipation of dimension reduction during the updating step.

Furthermore, another experiment on the same sequences is conducted to find out the reason why our *IMIMSL* is effective. As seen in Figure 2, the model with the parameters 4-3-3 (blue line) obviously outperforms the other two models, while the model with parameters 4-3-1 (green line) has slightly better results than the one with the parameters 1-1-1 (red line). The minor difference between the green line and the red line indicates that the use of multiple subspaces will enhance the learning ability of *IMIMSL*, but it is not the main reason why *IMIMSL* has better performance than other learning methods. The significant improvement of blue line makes it clear that the multiple samples combination strategy greatly increases the performance of subspace learning methods, which is not pointed out in previous literatures.

4. Discriminative Model with Improved Multiple Instance Boosting

Multiple Instance Boosting is a learning method in which the training samples are not individually treated. The probability of a sample to be positive is

$$p(z_t^d|x_t) \propto \exp\{\sum_i h_i\} \tag{5}$$

where z_t^d represents the observation of the improved MIBoosting model, and h_i is the selected weak classifier. The training samples are packed into bags and the optimization objective is the bag rather than the individual samples. Compared with the conventional online boosting learning methods, the multiple instance boosting method is more robust to occlusion. Please refer to [3, 20] for more details.

The key point in online boosting based methods is the way to update and estimate the weak classifier continuously after being given the samples. In the method proposed in [6, 3], the likelihood probability density function $p(x_j|y = 1)$ and $p(x_j|y = -1)$ are assumed to be the normal distribution which is not always true in practical, where x_j is the j^{th} dimension of the sample x. A more feasible way is to utilize the GMM instead of the single gaussian model to estimate the sample distribution $p(x_j|y = +1)$. According to the Bayesian rule, it is easy to get the continuous Bayesian weak regression function, that is $f_j(x) = \log \frac{p(x_j|y=+1)}{p(x_j|y=-1)}$ with the assumption that the positive and negative samples have the equal probability in task, namely $P(y = +1) = P(y = -1)$. When the weak classifier receives a positive bag with the samples $\{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}$, we calculate the mean value of the k^{th} dimension, namely $\bar{x}_k$. Then the mean value is utilized to compute the probability of each Gaussian model in GMM and the Gaussian model which gives out the largest probability is the most matched one. Then the matched Gaussian model is updated with the received samples according to the method mentioned in [3], while the unmatched ones will not be updated. The corresponding weights of Gaussian model are updated as $\omega = (1 - \lambda)\omega + \lambda M$, where $M = 1$ for the matched Gaussian model, $M = 0$ for the unmatched ones, and λ is the learning rate parameter which is set to be 0.2 in this paper. At the same time, the updating rules for the negative sample are similarly defined. Note that the form of weak classifier in our model remains the same as [3].

5. Experiments

We conduct some experiments to evaluate the performance of our joint model tracker. Our tracker is implemented in C++ code and runs at about 2 to 3 frames per second on the standard PC platform with 3.0GHz dual core CPU and 2GB memory without any optimization.

Experimental Setup The tracker is evaluated on 10 publicly available sequences which contains different challenging conditions. The sequences are issued in previous works: the sequences 1-3 from MIL[3], sequences 4-6 from VTD[9], sequences 7-8 from TLD[8] and sequences 9-10 from PROST[17]. Our tracker is initialized with the first frame and it outputs the trajectory of the target. The quantitative comparison results of IVT[10], FragTrack[1], SemiBoost[7], CoGD[21], MIL[3], PROST[17], VTD[9], TLD[8] and our tracker are shown in Figure 3, Table 1 and Table 2. More results can be found in the supplementary materials.

Parameters The search radius of the tracker is set to $[20, 50]$. For the generative model, we set 4 subspaces to represent the target, and select 3 nearest of them to estimate the probability of the candidate samples. The updating samples are collected in the circle region with radius 4 and every 4 representative samples are combined together to update the multiple subspaces. For the discriminative model, 3 Gaussian models are utilized for the positive and negative sample, the number of candidate haar-like features is set as 300 and about 50 of them are chosen to construct the classifier. For the positive bags, the samples are collected from the circle with the radius 8 and about 35 of the collected samples are packaged in a bag according to the weight assigned to them. For the negative bags, 65 samples are collected from the ring with the radius interval $[12, 40]$. Moreover, we utilize the default parameters of other trackers which are public available and choose the best one of 5 runs, or take the results directly from the published papers. Specifically, we reproduce the CoGD tracker in C++ code and adopt the parameters as described in [21].

MIL Sequences The sequences *tiger1* and *occlude2* present frequent occlusions for several times. MIL and VTD have relatively better performance, because MIL adopts multiple instance updating strategy which is very robust to occlusion, and VTD efficiently combines some specific appearances of the target. Meanwhile, since the supervision strategy increases the possibility of our tracker to precisely find the target, our tracker also has the good performance, as supported by Table 1 and Table 2. The heavy 360° appearance variation and the occlusion by other similar object always challenge the stability of the trackers, just like what happens in sequence *girl*. While small drift exists, VTD and our tracker have the best performance.

Seq.	JMT	IVT	CoGD	Semi	MIL	Frag	PROST	VTD	TLD
girl	**10.9**	40.4	14.1	22.8	31.6	25.4	19.0	12.8	35.7
occlude2	10.8	19.7	13.3	25.2	14.2	21.5	17.2	**9.4**	14.9
tiger1	**8.01**	80.7	29.7	14.4	8.35	29.3	-	22.3	12.6
animal	**6.71**	226	7.38	12.3	80.3	71.4	-	9.68	50.7
basketball	**7.46**	95.4	13.8	153	93.3	12.7	-	11	158
football	7.33	17.2	9.16	102	12.7	9.92	-	**6.25**	13.0
jumping	4.71	62.1	**3.75**	59.7	10.2	5.45	-	40.9	5.04
panda	**5.68**	58.2	64.5	41.7	9.42	6.85	-	6.32	17.7
lemming	**12.5**	128	39.8	99.8	40.5	82.8	25.1	98	167
board	**25.3**	169	74.5	389	69.2	90.1	39.0	70.1	134

Table 1. Comparison results of average error center location in pixel.

Seq.	*Frame*	JMT	IVT	CoGD	Semi	MIL	Frag	PROST	VTD	TLD
girl	**502**	492	353	482	388	378	378	447	**502**	219
occlude2	**812**	**812**	583	767	548	807	618	665	792	712
tiger1	**354**	265	35	170	224	**279**	155	-	189	150
animal	**71**	65	3	62	56	5	13	-	**66**	43
basketball	**725**	**715**	75	335	90	175	630	-	601	15
football	**362**	**357**	246	292	65	272	302	-	**357**	272
jumping	**313**	**313**	65	308	35	109	258	-	79	209
panda	**1000**	**645**	120	175	375	195	465	-	510	315
lemming	**1336**	**1117**	284	907	733	882	733	942	471	234
board	**698**	**618**	30	279	105	354	474	524	274	95

Table 2. Tracking results. The total frame number of the sequences are presented in the second column. The number in other columns indicate the count of successful tracking frames based on the evaluation metric of PASCAL VOC object detection[5].

VTD Sequences There are numerous objects similar to the target in the background of the sequences *animal*, *football* and *basketball*. As seen in Figure 3, these similar objects always distract the detection based trackers away such as TLD and SemiBoost, because the appearances of the similar objects and the target are visually undistinguishable. Thanks to the supervision of the two different models in the multiple instance way, our tracker produces a little better performance than VTD which is very good at dealing with this kind of challenges.

TLD Sequences The sequence *jumping* contains abrupt motion because of handhold camera, and motion blur resulting from quick motion. These problems can be handled with an efficient appearance model or a detection module, such as FragTrack and TLD. Through combining a generative model and a discriminative model, CoGD and our tracker have more satisfactory results than other trackers. The frequent non-grid appearance variations in sequence *panda* result in less accuracy in tracking results, for example, IVT, CoGD and SemiBoost almost lose the target during tracking procedure. Since VTD incorporates several basic observation models and motion models into a compound tracker, it performs well in this sequence, but its tracking performance is still not as satisfactory as ours, as seen in Table 1 and Table 2.

PROST Sequences The cluttered background in sequences *lemming* and *board* actually confuses the trackers a lot. Even the very stable tracker VTD easily loses the target, and TLD and SemiBoost based on detectors cannot successfully track the target for a long time, because the too much background information in bounding box leads to the failure of the detectors. Relatively, the trackers including CoGD, MIL and PROST which take the surroundings into account outperform other trackers including IVT and FragTrack which just consider appearance features. As illustrated in Figure 3, Table 1 and Table 2 experimentally, our tracker has the best performance because the joint model can effectively alleviate the influence of noise introduced by the complex background.

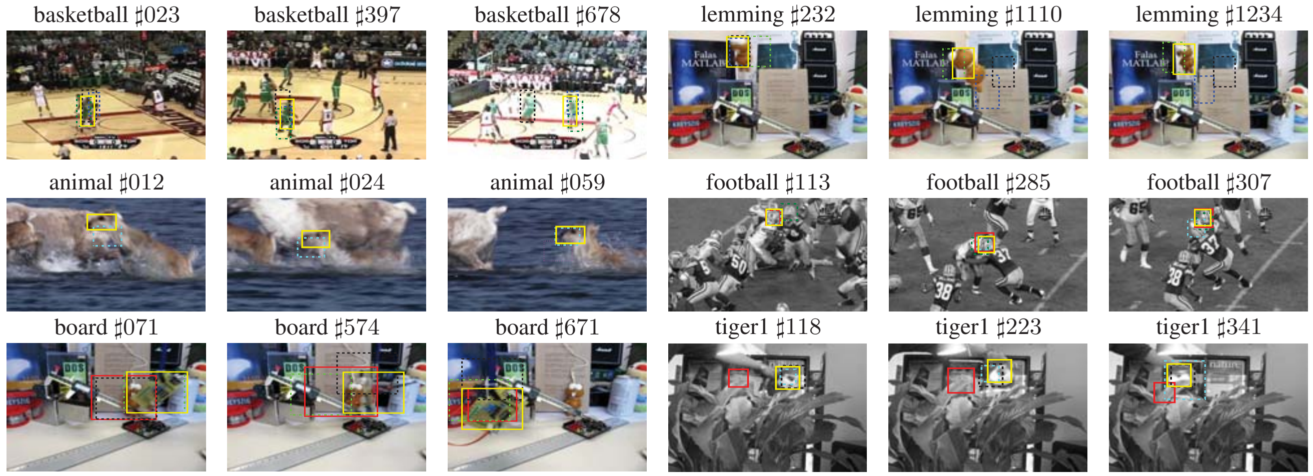

Figure 3. Tracking results. The results of our tracker, FragTrack[1], SemiBoost[7], CoGD[21], MIL[3], PROST[17], TLD[8] and VTD[9] are depicted as yellow, dark green, magenta, blue, black, light green, cyan and red rectangles respectively. Only the trackers with relatively better performances of each sequences are displayed.

6. Conclusion

In this paper, a multiple instance joint model based robust tracker is proposed. The target appearance is constructed using the *IMIMSL* model that learns the appearance variations of the target and the improved MIBoosting model that differentiates the target from the background. The two parts of the model provide updating samples for each other and they are updated in the multiple instance way. Experimental comparison with the state-of-the-art tracking strategies demonstrates the superiority of our joint tracker. Our future work includes the introduction of the adaptive weight between the *IMIMSL* model and the improved MIBoosting model, which will provide our tracker more robustness.

Acknowledgement

This work was supported by the Chinese National Natural Science Foundation Project #61070146, #61105023, #61103156, #61105037, National IoT R&D Project #2150510, Chinese Academy of Sciences Project No. KGZD-EW-102-2, European Union FP7 Project #257289 (TABULA RASA http://www.tabularasa-euproject.org), and AuthenMetric R&D Funds.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006. 1, 5, 6

[2] S. Avidan. Ensemble tracking. In *CVPR*, pages 494–501, 2005. 1

[3] B. Babenko, M.-H. Yang, and S. J. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009. 1, 2, 4, 5, 6

[4] P. Dollár, Z. Tu, H. Tao, and S. Belongie. Feature mining for image classification. In *CVPR*, 2007. 2

[5] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 5

[6] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, pages 260–267, 2006. 1, 4

[7] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247, 2008. 1, 5, 6

[8] Z. Kalal, J. Matas, and K. Mikolajczyk. P-N learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, pages 49–56, 2010. 1, 5, 6

[9] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010. 5, 6

[10] J. Lim, D. A. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In *NIPS*, 2004. 1, 2, 3, 4, 5

[11] B. Liu, J. Huang, L. Yang, and C. A. Kulikowski. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, pages 1313–1320, 2011. 1

[12] R. Liu, J. Cheng, and H. Lu. A robust boosting tracker with minimum error bound in a co-training framework. In *ICCV*, pages 1459–1466, 2009. 1

[13] H. Lu, Q. Zhou, D. Wang, and R. Xiang. A co-training framework for visual tracking with multiple instance learning. In *FG*, pages 539–544, 2011. 1

[14] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):696–710, 1997. 3

[15] H. T. Nguyen, Q. Ji, and A. W. M. Smeulders. Spatio-temporal context for robust multitarget tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):52–64, 2007. 4

[16] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *CVPR*, pages 728–735, 2006. 1

[17] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. PROST: Parallel robust online simple tracking. In *CVPR*, pages 723–730, 2010. 1, 5, 6

[18] F. Y. Shu Wang, Huchuan Lu and M.-H. Yang. Superpixel tracking. *ICCV*, 2011. 1

[19] M. Tipping and C. Bishop. Probabilistic principal component analysis. *J. Royal Statistical Soc. Series B*, 61(3):611–622, 1999. 2

[20] P. A. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005. 4

[21] Q. Yu, T. B. Dinh, and G. G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, pages 678–691, 2008. 1, 2, 3, 4, 5, 6