This ICCV2013 Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

An Adaptive Combination of Multiple Features for Robust Tracking in Real Scene

Weihua Chen Lijun Cao Junge Zhang Kaiqi Huang
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{weihua.chen, ljcao, jgzhang, kqhuang}@nlpr.ia.ac.cn

Abstract

Real scene video surveillance always involves low resolutions, lack of illumination or cluttered environments, which leads to insufficiency of discriminative details for the target. In this situation, discrimination based tracking methods could fail. To address this problem, this paper presents an adaptive multi-feature integration method in terms of feature invariance, which can evaluate the stability of features in sequential frames. The adaptive integrated feature (AIF) is consisted of several features with dynamic weights, which describe the degree of invariance of each single feature. An incremental principal component analysis (IPCA) adjusted by the accuracy of tracking results is used to update the adaptive integrated feature, and partially avoids the problem of "updating dilemma", which is common in most of adaptive updating methods. Experiments on pedestrian tracking demonstrate the proposed approach is effective and shows improved performance compared with several state-of-the-art methods in real surveillance scenes.

1. Introduction

Objecting tracking is a fundamental task in computer vision with wide applications, such as video surveillance and human computer interaction. As a useful visual tracking algorithm, it should be real-time and be designed to handle cases in unconstrained environments for a long duration. The common challenges are usually due to appearance changes, partial occlusions, cluttered backgrounds, and illumination changes.

As a classic tracker, to overcome these problems, it's usually researched from two aspects: representation and tracking strategy. Especially tracking strategy has become more and more popular. In [1], some of the best trackers [2, 3, 4] belong to this "tracking strategy" level. Most of them focus on a tracking by detection model with an adaptive learning process. TLD [2] gives good performance in many datasets because of its adaptive P-N learning based on a randomized forest classifier. Struck [3] proposes an improved SVM to rebuild the tracking framework, while MIL [4] offers an online multiple instance boosting to lead its tracker. Their works concentrate on "tracking strategy" level and lack improvements from representation level, which also play a major role in visual tracking and are potentially complementary for them. TLD's feature is only based on 2 bits LBP. Struck and MIL features are limited to Haar-like and raw features. They can't preform very well in a blur environment where the local features lose their edge, as shown in Fig.1. Besides the representation problem, the learning process causes the low computing speed. So these trackers are difficult to meet the real-time requirement. So this paper pays attention to representation level.

Since an appropriate appearance model plays a crucial role in visual tracking, this paper focuses on how to learn an effective appearance model at each frame. Traditional algorithms always split this process into two steps: feature extraction and modeling. However, in recent years, many "representation-level" trackers [5, 6, 7, 8, 9, 10] combines these two steps together. They use some machine learning methods to build their models, *e.g*. Sparse or PCA. So their inputs are just the raw feature (grayscale). Their good performance is due to the high efficiency of machine learning algorithms. In real scenes, affected by low resolution and lack of illumination, the target is always somewhat blurred and the raw feature is not as discriminative as wanted. Under this situation, using raw features directly is hard to get the valid information of the target. For this problem, the proposed method extracts high-level features, *e.g*. Color, HOG [11], LBP [12] and Haar [13], instead of raw feature and focus on the feature invariance as the valid information to represent the target.

To build appearance model, the key is to combine multiple high-level features together effectively and form a more suitable representation [14, 15, 16, 17], so that combination of multiple features can achieve more robust tracking than

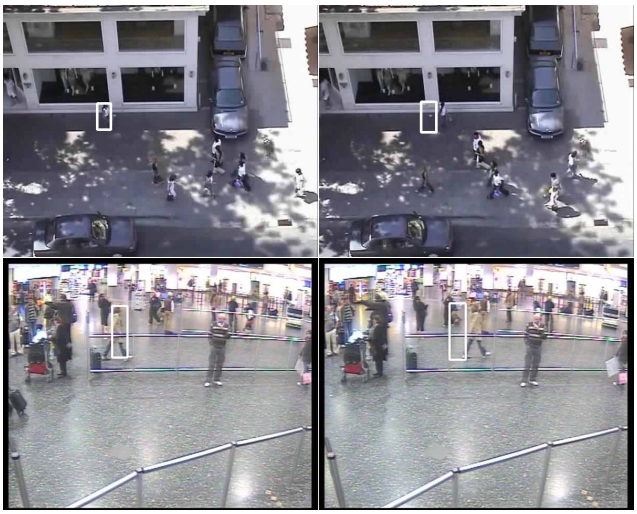

Figure 1. Some results of TLD Tracker.

any single feature. There are some important issues: how to combine multiple high-level features and how to determine the contribution of each feature. Collins *et al*. [14] uses a feature pool to select the distinctive feature from background. Kwon and Lee [15] integrate multiple features together and apply SPCA (Sparse Principal Component Analysis) to compute the dissimilarities among different models. These methods are robust to the bad drift problem by emphasizing the distinction property of features [18, 19, 20]. However, discrimination of objects can decrease for low resolutions, lack of illumination or cluttered environments in real scenes. In this case, distinction should not be the most useful information extracted from the high-level features. As a result, the proposed method concentrates on the feature invariance instead. This feature invariance provides a contribution for robust tracking in this "undistinctive" situation. Park *et al*. [21] and Jepson *et al*. [22] update the appearance model adaptively by invariance, which lead to a stable model for target estimation. In this paper, the invariance property of high-level features is adopted as a measure to represent the target and update their weights frame by frame.

An adaptive updating process is not only important but also necessary for a robust tracker. Many trackers [2, 10, 23, 24, 25, 26] use this adaptation to update the appearance model. But most of them with an adaptive model are likely to meet a problem of "updating dilemma". The reason is that current model is updated under the assumption that the previous model and tracking results are correct. Unfortunately, this assumption may not always hold because the interference always exists. So, due to this problem, a confidence value is brought in, which reflects the tracking accuracy to control the updating rate.

In general, this paper presents a new adaptive combination of multiple features in terms of feature invariance for robust tracking. Given a set of features, an adaptive integrated feature (AIF) is formed by combining of these features (denoted as Sub-f) with dynamic weights determined by the invariance degree of Sub-fs frame by frame. The Sub-f model is updated based on the accuracy of tracking results to tackle the "updating dilemma". As a result, the appearance model represented by AIF is also adaptively updated.

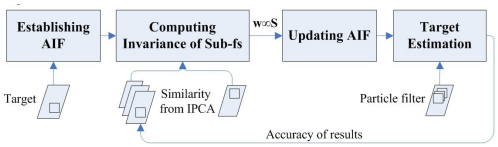

Figure 2. An illustration of the proposed method.

It is worth mentioning that in this paper, unlike other algorithms, Incremental principal component analysis (IPCA) is not used for a simple dimensionality reduction. The proposed method takes advantage of an IPCA process to establish a subspace (by its principal components), which is considered to represent the main structure of the target from previous frames. As a result, the corresponding eigenvalues can be seen as a description of the target previous state. To obtain the current state, the current frame is projected onto the main structure. The similarity between the current state and the previous state (the eigenvalues got from IPCA) represents the feature invariance. So the IPCA process plays a crucial role in computing the invariance of Sub-fs.

This paper contributes to the research of adaptive tracking in the following ways: (1) An adaptive integrated feature, which emphasizes the property of invariance, is proposed; (2) The updating process partly avoids the problem of "updating dilemma" by adaptively updating the Sub-f model according to the accuracy of tracking results; (3) This method makes real-time tracking in unconstrained environments for a long duration possible.

The rest of the paper is organized as follows. The AIF structure is described in Section 2.1. In Section 2.2 and 2.3, the invariance of each Sub-f is computed and AIF is updated respectively. The target is estimated in Section 2.4. The experimental results and conclusions are given in Section 3 and Section 4.

2. The proposed method

In this section the proposed method is presented in detail. First, the structure of the AIF is introduced. Next, it is explained that how to calculate the invariance of each Sub-f by IPCA. Then this invariance is applied to update the Sub-fs' weights. At last, a particle filter tracker is used to estimate the target and the Sub-fs are updated according to

the accuracy of tracking results.

An illustration of the proposed method is presented in Fig.2.

2.1. Establishing AIF

It is assumed that the Sub-fs are independent; therefore, the AIF is consisted of a series of the Sub-fs. The structure of AIF is listed as below,

$$AIF = [w_1 f_1, w_2 f_2, ..., w_n f_n] \tag{1}$$

where f_i is the Sub-f with a length of D, $f_i \subset R^{D*1}$, n is the number of Sub-fs, w_i is the weight, and $\sum_{i=1}^{n} w_i = 1$. All Sub-fs have the same dimension.

In the initialization, the default setting is $w_i = 1/n (i = 1, ..., n)$.

2.2. Computing invariance of sub-features

The invariance of Sub-f is the key to improve the AIF in this method. An IPCA process is used to compute the principal components of each Sub-f in previous frames, and get the subspace of basis vectors and the corresponding eigenvalues. Then the Sub-f of current frame is projected onto this subspace to obtain the vector of projection values. The correlation between eigenvalues and projection values is used to evaluate the invariance. The details are described as follows:

2.2.1 Initialization stage. In the beginning of tracking, there's not enough positive samples. As a result, an initialization stage is needed. In the first t frames, k samples are produced from every frame, and $Q = k * t$ samples are obtained as our positive samples. These samples X are produced by varying the target scales and rotations:

$$X = (x_1, ..., x_Q) \subset R^{D*Q} \tag{2}$$

Based on the training samples, the eigenvectors U of the Sub-fs can be easily got.

$$U = (u_1, ..., u_D) \subset R^{D*D} \tag{3}$$

2.2.2 Updating feature space. The covariance-free IPCA [27] is adopted to update the eigenvectors directly according to previous eigenvectors and a new observation image. It has the advantage of real time, and the IPCA process is shown as follows:

$$v(N) = (1 - \alpha) v(N-1) + \alpha x_N x_N^T \frac{v(N-1)}{\|v(N-1)\|} \tag{4}$$

where $v(N)$ is the updated eigenvector when the new sample x_N is added in. α is the updating rate, reflecting the accuracy of the new data (see Section 2.4).

The updating process is more reasonable through the adjusting of the rate α instead of a constant. In a constant updating, the model at the current time is updated by the current tracking result, but this result actually may be far from the right model, and thus deviating the adaptation and failing the tracker. Therefore, in order to mend this problem, constrains of the new data in updating need to be enforced. In Eq.4, the feature space updating is constrained by the accuracy of tracking, which decreases the deviation caused by bad results and partially avoids the problem of "updating dilemma".

Eq.4 is used to update the eigenvectors of Sub-f iteratively, and $v(0) = u$.

2.2.3 Extracting principal component. When obtaining a new eigenvector v, its corresponding eigenvalue $\lambda = \|v\|$ is also available. All the eigenvectors are ordered by their eigenvalues to find the first d principal components which compose the Sub-f's subspace V.

$$d = \arg\min_d (\sum_{i=1}^{d} \lambda_i / \sum_{i=1}^{D} \lambda_i > \beta) \tag{5}$$

where β is a threshold to obtain the principle components and set to 0.9. It is used to remove some noise in subspace V.

Then, the subspace V and the vector of corresponding eigenvalues Λ are obtained as follows:

$$\begin{aligned} V &= (u_1, ..., u_d) \subset R^{D*d} \\ \Lambda &= (\lambda_1, ..., \lambda_d)^T \subset R^{d*1} \end{aligned} \tag{6}$$

2.2.4 Obtaining invariance. When a new frame $t+1$ comes, the vector Γ of its projection values on the subspace V is

$$\begin{aligned} \Gamma &= V^T \cdot x_{t+1} = (u_1, ..., u_d)^T \cdot x_{t+1} \\ &= (\gamma_1, \gamma_2, ..., \gamma_d)^T \subset R^{d*1} \end{aligned} \tag{7}$$

The similarity $S^{(t+1)}$ between Γ and Λ is measured by the correlation coefficient.

$$S^{(t+1)}(\Lambda, \Gamma) = \frac{\Lambda^T \cdot \Gamma}{\|\Lambda\| \|\Gamma\|} \tag{8}$$

This correlation $S^{(t+1)}$ reflects the similarity between the pervious state and the current state of the target. It is assumed that the feature invariance means the ability of the feature to keep the original state. This similarity indicates how many stable components exist in both previous and current states. And as the same idea of Jepson *et al*. [22], these stable components are exactly the invariance of the feature.

2.3. AIF updating

w, as the weight of Sub-f in AIF, is updated frame by frame. It reflects the invariance of Sub-f in the current frame, which means the Sub-f with a higher w is more stable. As a result, AIF not only covers all the Sub-fs' ability of

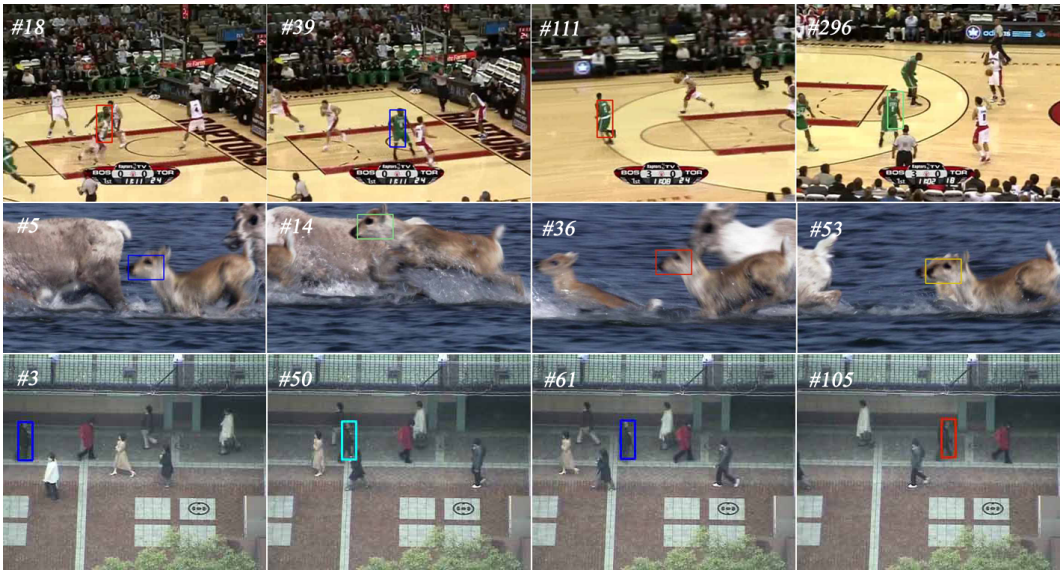

Figure 3. Some key frames in 1st set of experiments. The most invariant Sub-f: Red-Color, Blue-HOG, Green-Raw, Sapphire-LBP, Yellow-Haar.

representation, but also provides a more robust description than any Sub-f or their simple combination. w is updated as follows:

$$\begin{aligned} w'^{(t+1)}_i &= w'^{(t)}_i + \eta(S^{(t+1)}_i - S^{(t)}_i) \\ w^{(t+1)}_i &= w'^{(t+1)}_i / \textstyle\sum_{i=1}^{n} w'^{(t+1)}_i \end{aligned} \tag{9}$$

where η is the updating rate of w, and $w^{(t+1)}_i$ is the normalized weight of the ith Sub-f. Through the updated w, the more invariant Sub-f is more predominant in AIF.

When a target comes into view with strong interference on a certain Sub-f, the invariance of the Sub-f decreases. With the updating of w, the more robust Sub-f takes a higher ratio and the invariance of AIF is also enhanced for robust tracking.

2.4. Target estimation

In this stage, AIF is embedded in a particle filter tracking system [28], which uses a set of observations to estimate the true target position, and obtains the tracking results. To tackle the problem of "updating dilemma", the correlation of current AIF with the set of observations is used to evaluate the accuracy of tracking results and update the feature space of Sub-f. It's an average similarity between current target and all the observations. The only different in this paper from [28] is that the feature used in particle filter tracking in this paper is the variable AIF. So the updating rate α in Eq.4 can be figured out via Eq.10 below:

$$\alpha \propto (\sum_{j=1}^{m} \frac{O_j^T \cdot \hat{O}}{\|O_j\|\|\hat{O}\|})/m \tag{10}$$

where O is the set of observations, and $\hat{O}$ is the current AIF, m is the number of observations used in particle filtering.

3. Experimental results

The experiments in this paper are conducted to three sets of experiments. The first two sets are designed to test the AIF tracker performance vs the single Sub-fs and some state-of-the-art methods in existed public sequences, which well represent the various visual attributes such as occlusion, illumination variation, motion blur and fast motion. The third set of experiments is tested on pedestrian tracking in real scenes with two selected trackers from the second experiment set. The pedestrian tracking is difficult but a crucial problem in surveillance, for there are many complex situations in real scenes, such as illumination changes, cluttered environment or low resolution. The head-shoulder part is taken as target instead of the whole body, which can partly avoid occlusions, and the whole body rectangle in figures is for clarity.

In experiments, gradient descriptor (HOG), texture descriptors (8bits LBP and Haar) and color descriptor (Color histogram) are integrated as Sub-fs. Each Sub-f has a length of 256, and the length of AIF is 4*256. The updating rate of weights η is set to 10 experientially. In the initialization stage, all the weights of Sub-fs are set to 1. And the first twenty frames are used for initialization and 25 samples (5 scales $\{0.25, 0.5, 1, 2, 4\}$ and 5 rotations $\{-10°, -5°, 0°, 5°, 10°\}$) are extracted from every frame. In tracking, object scale is fixed according to the scene prior. All programs are implemented in C++ on a PC with Intel i7 3770 CPU (3.4GHz).

	AIF	SSC	HOG	Color	LBP	Raw	Haar
Basketball	**0.834483**	0.430345	0.073103	**0.711724**	0.464828	0.464828	0.085517
David	**0.996994**	**0.996782**	0.960894	0.199255	0.986854	0.991723	0.309125
Deer	0.577465	**0.605634**	0.323944	0.535211	0.422535	**0.591549**	0.28169
FaceOcc	**0.762332**	**0.794843**	0.760058	0.280269	0.700673	0.752242	0.457399
Football	**0.944751**	0.198895	**0.966851**	0.063536	0.69337	0.160221	0.544199
Jumping	**0.990671**	**0.933866**	0.108626	0.242812	0.927476	0.242812	0.111821
Singer	**0.629508**	**0.491257**	0.114754	0.135519	0.425683	0.379235	0.330055
Subway	**0.942857**	0.628571	**0.817143**	0.091429	0.480000	0.262857	0.388571
Sylvester	**0.932342**	0.884758	**0.977695**	0.113755	0.687732	0.863941	0.160595
Average	**0.845711**	**0.662772**	0.567007	0.263723	0.643239	0.523267	0.296552

Table 1. Precision plots of TRE in 1st set of experiments. Location error threshold = 20. Bold font means the two best scores.

	AIF	SSC	HOG	Color	LBP	Raw	Haar
Basketball	**0.720000**	0.354483	0.060690	**0.689655**	0.342069	0.451034	0.081379
David	0.750466	0.659218	**0.960894**	0.076350	0.456238	**0.968343**	0.301676
Deer	0.535211	**0.577465**	0.338028	0.535211	0.408451	**0.591549**	0.267606
FaceOcc	**0.982063**	**0.993274**	0.913677	0.533632	0.942825	0.961883	0.503363
Football	**0.883978**	0.187845	**0.944751**	0.022099	0.466851	0.160221	0.500000
Jumping	**0.801597**	**0.739936**	0.605431	0.163898	0.704792	0.240575	0.602236
Singer	**0.567760**	**0.537705**	0.122951	0.046448	0.426776	0.081967	0.030055
Subway	**0.662857**	0.411429	**0.588571**	0.062857	0.451429	0.234286	0.382857
Sylvester	0.809665	0.765799	**0.926394**	0.065428	0.523420	**0.813383**	0.154647
Average	**0.745955**	**0.580794**	0.584598	0.243953	0.524761	0.500360	0.313757

Table 2. Success plots of TRE in 1st set of experiments. Overlap threshold = 0.5. Bold font means the two best scores.

Both of the first two experiments contain the same 9 sequences from [1]. The performances are referred as spatial robustness evaluation (TRE) [1] from two aspects (precision plot and success plot). As the representative precision score for each tracker, the threshold for the score is set to 20 pixels. And in success plot, the representative threshold for tracker evaluation is $t_0 = 0.5$. In the first set of experiments, the AIF is compared with the single Sub-f and a simple series combination (SSC) of Sub-fs respectively under a framework of particle filtering. As shown in Table.1 and Table.2, the SSC tracker gives a better performance than each single Sub-f, but still has a certain gap compared with the proposed AIF tracker. Fig.3 offers some key frames where the switch of the main Sub-f happens. The color of head-shoulder rectangle indicates the most invariant Sub-f in current frame (red-Color, blue-HOG, green-Raw, sapphire-LBP, yellow-Haar). From Sequence Basketball (1st row) and Sequence Subway (3rd row) of Fig.3, it is obvious that the HOG and Color features play a major role in human tracking, especially under a distinctive environment. But when the similar-color target approaches (Sequence Deer ♯14 and ♯53; Sequence Basketball ♯296; Sequence Subway ♯50), the AIF switch the main Sub-f to local feature (such as LBP or Haar) to avoid the bad drift.

	AIF	TLD	Struck	MIL	VTS	Frag
FPS-A	31.6	28.1	20.2	39.1	5.7	6.3

Table 5. Average speed in 2nd set of experiments. FPS-A:average FPS.

What's more, to an occlusion problem, the AIF can partly handle it by adjust feature weights. As known the occlusion always happens in a temporary rush, which may cause a rapid change in features. The aim of AIF is to find the most invariant feature with the least change, so this feature will take a leading role in tracking to reduce the impact of occlusions. Some occlusion results are shown in Sequence Basketball ♯18 and Sequence Subway ♯50 in Fig.3.

In the second set, TLD tracker [2], Struck tracker [3], MIL tracker [4], Frag tracker [29] and VTS tracker [15] are cited to compare with the AIF tracker. The precision plot and success plot of the performances are shown in Table.3 and Table.4. VTS is a recent method using multi-sample, not only features but also image blocks, to search the appropriate trackers in each frame. So its representation can be considered as an advanced feature selection mixed with blocks. The comparison with VTS shows the effectiveness and robustness of the AIF tracker. TLD, Struck and MIL

	AIF	TLD[2]	Struck[3]	MIL[4]	VTS[15]	Frag[29]
Basketball	**0.834483**	0.366676	0.466437	0.595516	**0.969488**	0.687583
David	0.996994	0.984973	**0.998748**	0.941503	**1.000000**	0.844902
Deer	0.577465	0.581720	**1.000000**	0.453763	0.133333	**0.683871**
FaceOcc	0.762332	0.427423	**0.918596**	0.400255	0.705669	**0.981954**
Football	0.944751	0.547917	0.854167	**0.971875**	**0.989583**	0.931250
Jumping	0.990671	0.954925	**1.000000**	0.933433	0.397910	**0.996418**
Singer	**0.629508**	0.067010	0.609536	0.543557	**0.849227**	0.266753
Subway	**0.942857**	0.717187	0.570313	**0.856250**	0.453646	0.809896
Sylvester	0.932342	**0.936430**	**0.992392**	0.747623	0.746672	0.679298
Average	**0.845711**	0.620473	**0.823354**	0.715975	0.693947	0.764658

Table 3. Precision plots of TRE in 2nd set of experiments. Location error threshold = 20. Bold font means the two best scores.

	AIF	TLD[2]	Struck[3]	MIL[4]	VTS[15]	Frag[29]
Basketball	**0.720000**	0.253914	0.375696	0.509154	**0.871717**	0.572433
David	0.750466	0.824329	**0.985510**	0.440072	**0.918605**	0.801431
Deer	0.535211	0.579570	**1.000000**	0.451613	0.107527	**0.675269**
FaceOcc	0.982063	0.801883	**0.999902**	0.801491	0.895155	**0.999510**
Football	**0.883978**	0.483333	0.697917	0.606250	0.573958	**0.756250**
Jumping	0.801597	0.796716	**0.901791**	0.684776	0.217612	**0.942388**
Singer	0.567760	0.067268	**0.609278**	0.593299	**0.880412**	0.296907
Subway	0.662857	0.657292	0.535417	**0.733333**	0.409375	**0.719271**
Sylvester	**0.809665**	0.783102	**0.919971**	0.578713	0.692173	0.588661
Average	**0.745955**	0.583045	**0.780609**	0.599855	0.618503	0.705791

Table 4. Success plots of TRE in 2nd set of experiments. Overlap threshold = 0.5. Bold font means the two best scores.

	Color	HOG	LBP	SSC	AIF
frames	102	247	189	311	455

Table 6. Average valid frames compared with each Sub-f in 3rd set of experiments

	AIF	TLD	VTS
frames	455	317	287

Table 7. Average valid frames compared with other trackers in 3rd set of experiments

are some of the best trackers so far in [1]. As mentioned before, their works emphasize improving the tracker strategy, a detection tracking or an improved learning framework instead of feature extraction and representation. While the proposed method focuses on feature-level. So the proposed approach is somewhat complementary for them. That also explains why the AIF tracker is not as good as Struck in some specific sequences, such as Sequence Deer. Because in this sequence, the local feature is more robustness than other general features. The average FPS of tracking speed is shown in Table.5. The purpose of these comparisons is to show a good performance even vs the best trackers in visual tracking.

In the third set of experiments, 30 video sequences are collected from the real surveillance video under different scenes, the total pedestrians used in tracking is more than 100. The performance of trackers is evaluated by the average valid tracking frames (tracking result having over 50% overlap with the ground truth is taken as valid). The reason why this set of experiments doesn't choose the tracking accuracy (compare with ground truth) as the evaluation method is that success or lose of target tracking is more concerned than the accuracy of tracking in surveillance in real scenes. The proposed method is still compared with each Sub-f and some state-of-the-art methods (TLD and VTS) in some difficult conditions, i.e, appearance changes (people turning round) and drastic illumination changes. The results are listed in Table.6 and Table.7, from which it can be seen that the AIF tracker can follow the target much longer than others. In the rest of this section, several typical sequences are selected to present for illustration.

One of the sequences in the 3rd set is shown in Fig.4, which contains a person turning round and an intense interference in background. The LBP tracker loses target in frame ♯66 for a similar texture arising nearby and the SSC tracker fails in frame ♯268 for the person turning round in

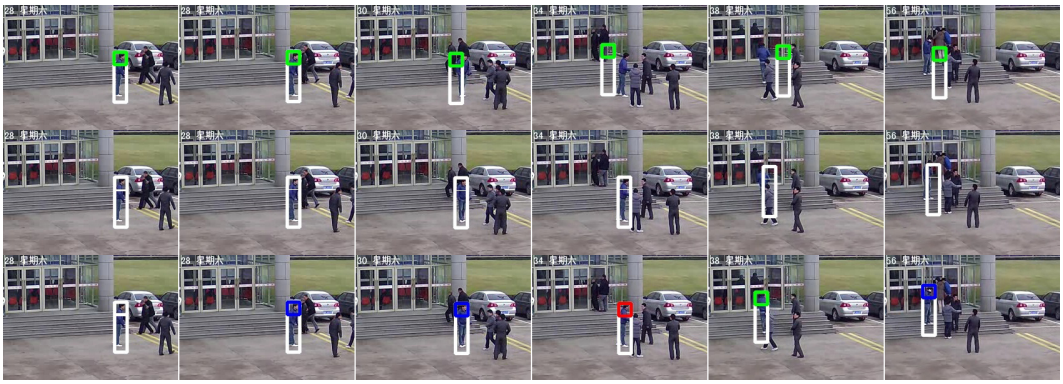

Figure 4. Tracking for frame ♯1,21,66,170,268 and 704, (1st row) LBP tracker, (2nd row) SSC tracker, and (3rd row) AIF tracker.

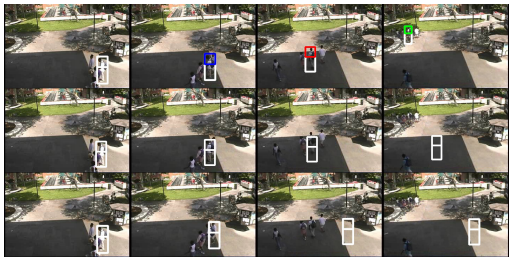

Figure 5. Tracking for frame ♯1, 41, 148 and 429, (1st row) AIF tracker, (2nd row) TLD tracker, and (3rd row) VTS tracker.

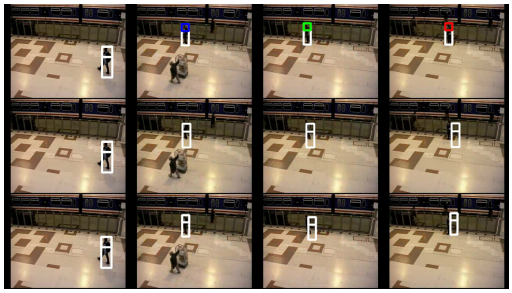

Figure 6. Tracking for frame ♯1, 116, 191 and 391, (1st row) AIF tracker, (2nd row) TLD tracker, and (3rd row) VTS tracker.

cluttered background. However, the AIF tracker can handle these problems effectively. From the 3rd row of Fig.4, it's clear that the AIF tracker has a better performance. It switches to color feature adaptively to follow target during the person turning round in frame ♯170 to overcome the appearance changes. And while an intense gradient interference coming in frame ♯268, the tracker changes to LBP as a major feature instead of HOG dynamically where the SSC tracker fails.

Fig.5 and Fig.6 are part results of the third set. Fig.5, the results of VTS lost target in frame ♯41 when the target gets into the shadow. TLD also fails in frame ♯148 during the person turning round. The appearance changes and the high gradient interference from drastic illumination changes caused these failures. The AIF tracker effectively track the target till the target goes out of the scene. Fig.6 is a person turning round frequently. TLD and VTS separately fails in frame ♯116 and ♯191. The proposed method overcomes it with a dynamic adjustment of different Sub-fs. At the beginning, the HOG descriptor is the main feature for tracking. Then the LBP descriptor arises instead of HOG gradually when the target goes into a high gradient background. During the person turning round, both HOG and LBP become instable and AIF switches to Color descriptor adaptively for robust tracking.

4. Conclusions

A novel object tracking algorithm based on an adaptive integrated feature is proposed. It integrates features in terms of invariance and establishes an adaptive target representation for robust tracking in low quality real scenes. An IPCA process is utilized to find the invariance of subfeatures for time efficiency, and is updated by the accuracy of tracking results to solve the problem of "updating dilemma" in adaptive tracking. This algorithm is general, robust and computationally efficient. Experiments on challenging datasets show that AIF tracker can handle complex appearance changes of objects and camouflage environments. The future work is to improve the AIF to handle the occlusions of multiple objects.

5. Acknowledgement

This work is funded by the National Basic Research Program of China (Grant No. 2012CB316302), the National Key Technology R&D Program (Grant No. 2012BAH07B01), and the National Natural Science Foundation of China (61105009).

References

[1] Y. Ming, J. Yuan, and M. Yang, "Online object tracking: a benchmark," in *Computer Vision and Pattern Recognition*, IEEE, 2013.

[2] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence*, vol. 34(7), pp. 1409–1422, July 2012.

[3] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *International Conference on Computer Vision*, pp. 263–270, IEEE, 2011.

[4] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence*, vol. 33(8), pp. 1619–1632, August 2011.

[5] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer Vision and Pattern Recognition*, pp. 1822–1829, IEEE, 2012.

[6] W. Zhong, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer Vision and Pattern Recognition*, pp. 1838–1845, IEEE, 2012.

[7] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Computer Vision and Pattern Recognition*, pp. 2042–2049, IEEE, 2012.

[8] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition*, pp. 1830–1837, IEEE, 2012.

[9] J. Kwon and K. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition*, pp. 1269–1276, IEEE, 2010.

[10] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77(1-3), pp. 125–141, May 2008.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.

[12] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence*, vol. 24(7), pp. 971–987, July 2002.

[13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, IEEE, 2001.

[14] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence*, vol. 27(10), pp. 1631–1643, October 2005.

[15] J. Kwon and K. Lee, "Tracking by sampling trackers," in *International Conference on Computer Vision*, pp. 1195–1202, IEEE, 2011.

[16] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *European Conference on Computer Vision*, vol. 6314, pp. 624–637, IEEE, 2010.

[17] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Computer Vision and Pattern Recognition*, pp. 1815–1821, IEEE, 2012.

[18] W. Zhang, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer Vision and Pattern Recognition*, pp. 1838–1845, IEEE, 2012.

[19] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in *Computer Vision and Pattern Recognition*, pp. 2034–2041, IEEE, 2012.

[20] Y. Wu, J. Lim, and Y. Wu, "Spatial selection for attentional visual tracking," in *Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[21] D. Park, J. Kwon, and K. Lee, "Robust visual tracking using autoregressive hidden markov model," in *Computer Vision and Pattern Recognition*, pp. 1964–1971, IEEE, 2012.

[22] A. Jepson, D. Fleet, and T. EI-Maraghi, "Robust online appearance models for visual tracking," *Pattern Analysis and Machine Intelligence*, vol. 25(10), pp. 1296–1311, October 2003.

[23] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *European Conference on Computer Vision*, vol. 5302, pp. 234–247, IEEE, 2008.

[24] H. Lim, V. Morariu, O. Camps, and M. Sznaier, "Dynamica appearance modeling for human tracking," in *Computer Vision and Pattern Recognition*, vol. 1, pp. 751–757, IEEE, 2006.

[25] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: tracking should be as simple as detection, but not simpler than recognition," in *International Conference on Computer Vision Workshops*, pp. 1409–1416, IEEE, 2009.

[26] S. Avidan, "Ensemble tracking," *Pattern Analysis and Machine Intelligence*, vol. 29(2), pp. 261–271, February 2007.

[27] J. Weng, Y. Zhang, and W. Hwang, "Candid covariance-free incremental principal component analysis," *Pattern Analysis and Machine Intelligence*, vol. 25(8), pp. 1034–1040, August 2003.

[28] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29(1), pp. 5–28, August 1998.

[29] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Computer Vision and Pattern Recognition*, pp. 798–805, IEEE, 2006.