2015 IEEE Winter Conference on Applications of Computer Vision

Adaptive Local Movement Modelling for Object Tracking

Baochang Zhang*,† Zhigang Li† Alessandro Perina* Alessio Del Bue* Vittorio Murino*

* Pattern Analysis and Computer Vision (PAVIS)
Istituto Italiano di Tecnologia, Genova, Italy

† School of Automation Science and Electrical Engineering
Beihang University, Beijing, China

Abstract

In this paper we present a novel strategy for modelling the motion of local patches for single object tracking that can be seamlessly applied to most part-based trackers in the literature. The proposed Adaptive Local Movement Modelling (ALMM) method is able to model the local spatial distribution of the image patches defining the object to track and the reliability of each image patch. Given the output of a base tracking algorithm, a Gaussian Mixture Model (GMM) is first used to model the distribution of the movement of local patches relative to the gravity center of the tracked object. Then, the GMM is combined with the base tracker in a boosting framework, which gives a novel integrated boosting classifier for the tracking task. This provides a robust procedure to detect outliers in the local motion of the patches. The algorithm is highly configurable with the possibility to change the number of local patches used for tracking and to adapt to the variations of the tracked object. Tracking results on standard datasets show that equipping state-of-the-art trackers with our tehcnique remarkably improves their performance.

1. Introduction

Visual object tracking is important in most existing Computer Vision systems[1, 3]. Recent and much progresses in object tracking was yielded, but designing a robust tracker to track the objects of partially occluded and deformable targets is still a big challenge. In particular object representation and motion modelling are two main issues in visual tracking.

To address the appearance variation problems, researchers have developed many appearance models and corresponding methods for adapting these models during tracking. Noticeably, several methods based on discriminant features have been proposed recently [4, 5, 6]. Learning-based methods are one of the most promising ways in dealing with these tracking issues, since they can adapt to the rigid and non-rigid object texture and shape variations. Other works [7, 8] use a discriminative classifier to distinguish the object from the background.

Structured learning has shown good results on tracking a whole target, but its complexity increases when modelling the relationships between patches. In [4], a method that models unknown parts is proposed to predict the location of the specific regions with latent part variables. Despite many advances made in this area, deformable targets and partial occlusions are still key problems in visual tracking [21] [16][2].

The other major issue in the tracking task is the motion modelling of the single object position and its parts defined as a set of image patches. Probabilistic estimation frameworks, such as the Kalman or particle filters, are consolidated techniques for motion prediction [3]; unfortunately, the methods are widely known for suffering the drifting problem.

On the other hand, data association techniques [10, 11, 12] can find the motion pattern based on the detection information. Another probabilistic approach [13] exploits displacements as a distribution matrix, and introduces a filter and related tools that work directly on the matrix. In [10], the authors propose to extract discriminative features and associate similar trajectories to detect individual moving objects. In this method, possible motion inconsistencies between different body parts may cause wrong object tracking. The method proposed in [14] learns a single motion distribution at each frame location from videos of a urban traffic. The work in [15] also uses similar usage of local motion patterns to improve the tracking performance.

From the literature review, one can notice a trend emerging where trackers tend to become heavily dependent on a complicated learning methods; "*Is this really necessary?*", "*Can complex learning methods employed in real time ap-*

978-1-4799-6683-7/15 $31.00 © 2015 IEEE
DOI 10.1109/WACV.2015.11

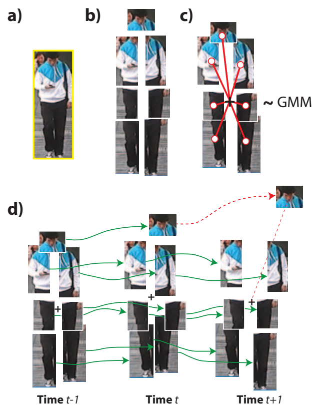

Figure 1. Overview of the proposed approach (ALMM): The approach uses as input a set of image patches, b) extracted from a detection a). The assumption of ALMM is that displacement of patches wrt the gravity center, follows a Gaussian mixture model (c). The proposed ALMM method gives more weights to the more robust patches given the output of the tracker and the local motion model based on GMM.

plications?". In this paper we follow the opposite direction and show that modelling the simple intrinsic local motion can significantly improve the tracking performance. In particular, with local movement we refer to the motion of a local patch center relative to the gravity center of the whole tracked object as llustrated by Fig. 1a-c. The basic idea of our approach is that the distribution of the movements of the local patches is easier to model instead of the whole object motion. Therefore first, we first calculated this distribution and subsequently we use it to verify whether a local patch is coming from the tracked object or not. This last step is called as the detecting outlier patches stage.

This paper directly addresses the local motion modelling problem and is proposes an Adaptive Local Movement Modelling (ALMM) method that includes a classifier which considers both object representation and local motion in a unified way. Fig. 1 shows a sketch of the idea underlying our approach. Given an input video sequence with the target object initialization including the corresponding image patches, we first perform tracking of the local patches using a *base tracker* (e.g., any off-the-shelf tracking algorithm would work and in this paper we considered [17, 19]) to calculate the position and the validity of each image patch. As second step, locations of the patches are *further* corrected using an outlier detection process based on a Gaussian Mixture Model (GMM)[20,22] that prunes the patches that diverge from the current statistics. Finally, after the GMM fitting, we assign a weight to each patch which is then used, together with the base tracker output, to decide whether a patch should be kept or discarded in computing the gravity center.

The rest of the paper is organized as follows. In Section 2, we introduce local patch movement modelling based on Gaussian mixture models. The adaptive local movement modelling based on the integrated boosting classifier is described in Section 3. Experimental results are reported in Section 4. Finally, we conclude the paper also envisaging the future work in Section 5.

2. Local patch movements using Gaussian mixture models

The traditional trackers are generally based on learning new object models using the image texture and this suffers from failures when the texture information is severely changed or even missing. In these situations, the local relative position information can be used to check whether the tracked result is correct. To fully exploit the local movement information in a tracking problem, we must solve the following issues:

1. Calculate the gravity center $\mathbf{G}$ of the whole tracked object based on the relative positions of local patches, also when only partial information is available.
2. Model the local movement of the patches in an online framework with a Gaussian Mixture model

Gravity center calculation based on local patches. We define the positions of the N local patches onto the image plane at frame t as $\mathbf{z}_1^t, \ldots, \mathbf{z}_n^t, \ldots \mathbf{z}_N^t$ where each $\mathbf{z}_n^t$ is a 2 dimensional x,y vector $\mathbf{z}_n^t = [z_{x,n}^t, z_{y,n}^t]$ computing taking the lower-left point of the detection window as origin as illustrated by Fig.2. The global gravity center position $\mathbf{G}^t$ for the t^{th} frame is then estimated as:

$$\begin{aligned} G_x^t &= \frac{1}{\sum_n v_n^{t-1}} \cdot \sum_{n=1}^{N} v_n^{t-1} \cdot \frac{G_x^{t-1}}{z_{x,n}^{t-1}} \cdot z_{x,n}^t \\ G_y^t &= \frac{1}{\sum_n v_n^{t-1}} \cdot \sum_{n=1}^{N} v_n^{t-1} \cdot \frac{G_y^{t-1}}{z_{y,n}^{t-1}} \cdot z_{y,n}^t \end{aligned} \quad (1)$$

In the last equation, v_n^{t-1} is a validity flag equal to 1 or 0 indicating respectively if a patch was considered valid or noisy and suppressed by our framework; the decision is taken by a boosting framework as explained later in the text. The initial position $\mathbf{G}^0$ is defined during the initialization stage, which lasts about 15 frames and the validity flags v_n^0 are initalized as 1.

Local patches position information. We modeled displacement of the local image patches from the gravity center with a Gaussian mixture model (GMM) of K components of means μ_k, standard deviations σ_k and mixing coefficients π_k.
At each frame time, patch and center, we compute $\mathcal{G}(k^t|\mathbf{z}_n^t)$

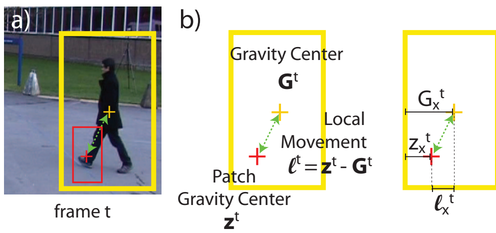

Figure 2. Local movement and gravity center illustration. a) A video frame with a tracked pedestrian and a patch. b) Local movement vector ℓ^t.

equal to 1 if the current sample falls inside 2.5 standard deviations σ_k^t from the mean μ_k and 0 otherwise. Then we update mean and variance of the GMM as follows

$$\begin{aligned}\mu_{k,x}^t &= (1-\rho)\cdot\mu_{k,x}^{t-1} \\ &+\rho\cdot\frac{1}{\sum_n \mathcal{G}(k^t|\mathbf{z}_n^t)}\cdot\sum_n^N \mathcal{G}(k^t|\mathbf{z}_n^t)\cdot\ell_{x,n}^t\end{aligned} \quad (2)$$

$$\begin{aligned}\sigma_{k,x}^{2,\ t} &= (1-\rho)\cdot\sigma_{k,x}^{2,\ t-1}+ \\ &+\rho\cdot\frac{1}{\sum_n \mathcal{G}(k^t|\mathbf{z}_n^t)}\cdot\sum_n^N \mathcal{G}(k^t|\mathbf{z}_n^t)\cdot(\ell_{n,x}^t-\mu_{k,x}^t)^2\end{aligned} \quad (3)$$

Similarly for the y-coordinate. The values $\ell_{n,x}^t$ represent the displacements, i.e., the current relative positions of the patches relative to the gravity center such that

$$\ell_{x,n}^t = z_{x,n}^t - G_x^t \quad \text{and} \quad \ell_{y,n}^t = z_{y,n}^t - G_y^t$$

as illustrated by Fig. 2a-b.
To make the calculation more efficient, the above process is implemented along x, y directions, individually.
Finally, each weight π_k^t of the k^{th} Gaussian component at the time t is obtained as:

$$\pi_k^t = (1-\rho)\cdot\pi_k^{t-1} \;+\; \rho\cdot\frac{\sum_n^N \mathcal{G}(k^t|\mathbf{z}_n^t)}{N} \quad (4)$$

In our experiments we set $\rho = 0.1$, which acts as a learning rate.
Once we have fit the GMM using Eqs. 2-4, we can detect outlier patches by using the following decision function $\mathcal{D}$, which is similar to what already proposed in [20]

$$\mathcal{D}(\mathbf{z}^t) = \begin{cases} 1 \;\; (\textit{local movement}) & \text{if } \sum_k \pi_k^t\cdot\mathcal{N}_k(\mathbf{z}^t) > 0.7 \\ 0 \;\; (\textit{outlier}) & \text{otherwise}\end{cases} \quad (5)$$

where $\mathcal{N}_k(\mathbf{z})$ is equal to 1 if the current sample falls inside 2.5 standard deviations σ from the mean, and 0 otherwise.
If $\mathcal{D}(\mathbf{z}^t) = 1$, we can consider that the local patch $\mathbf{z}^t$ comes from from the tracked object, otherwise we consider it as an outlier, as illustrated by Fig.3 where the cyan patches shown on the two frames on the right of the figures are considered outliers.

3. The integrated boosting classifier (ALMM)

After defining the local patches motion, we report here how to include the tracker information in an integrated boosting classifier.
In practice, the output of the base tracker and the outlier detection process aforementioned (Eq. 5), are both used to calculate a weight w_n^t of each local patch, which is then combined together into a decision stump classifier; while the base tracker decides if the given patch has a stable appearance in terms of texture, the GMM analyzes it in terms of displacement with respect to the gravity center.
The aim of this step is to "promote" patches that are stable in time because of good behavior in both motion and appearance and used them for a better estimation of the gravity center. The patches are weighted by errors of weak learners to promote patches which are correctly classified.

We define $\mathcal{B}(\mathbf{z}^t)$ as the output of the base tracker, which is equal to 1 if it has given a successful target position that falls inside the given searching region, and 0 otherwise. Again, it is important to note that any part-based tracker can be used.
Now, let w_n^t be the weight for the n^{th} patch at the iteration t. Similariy to boosting we initialize the weights as:

$$w_n^0 = \frac{1}{N} \qquad n = 1,\ldots,N. \quad (6)$$

The classification performance is actually a weighted sum of a local patch classifier at the iteration t. To update the weights, we first define H to be equal to

$$H(\mathbf{z}^t) = \begin{cases} 1 & \text{if } \;\mathcal{B}(\mathbf{z}^t)\wedge\mathcal{D}(\mathbf{z}^t) \\ -1 & \text{otherwise}\end{cases} \quad (7)$$

The value $H(\mathbf{z})$ is decided on the basis of both the base tracker outcome, i.e., if it has given a successful target position in the next frame, and the GMM fitting, i.e., if the target undergoes a good fit given the current GMM configuration, as defined in the previous section.
Then we define the error associated

$$\epsilon_n^t = \sum_{t'<t} w_n^{t'}\cdot[H(\mathbf{z}_n^{t'}) = -1] \quad (8)$$

where ϵ_n is the statistical error of the n^{th} patch and $[\cdot]$ is the indicator function, equal to 1 if the condition is true. Eventually the sum in Eq.8 can be windowed, considering only the previous T frames although experimentally we did not appreciated any change in perofrmance. Finally It is worth noticing that Eq. 8 can be efficiently computed recursively summing the error ϵ at time $t-1$ and an error term at time t, e.g., $\epsilon_n^t = \epsilon_n^{t-1} + e^t$

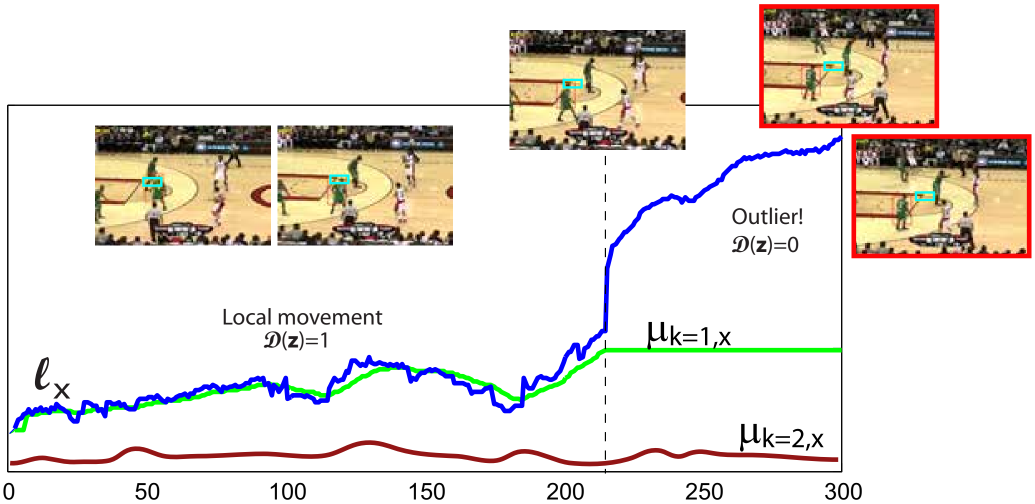

Figure 3. Modeling the patch misplacement with a Gaussian mixture model. The blue line represents $\ell_{n,x}^t$ for the cyan patch shown in the exemplar frames. Green and Red lines are the means of two centers of the GMM across the frame. After around frame 220, the patch diverge and it is not used to deremine the update the position of the gravity center.

Each candidate window in the tracking procedure is considered in the above boosting process such that:

$$w_n^t = \frac{w_n^{t-1} \cdot e^{-\alpha^{t-1} \cdot H(\mathbf{z}_n^{t-1})}}{\sum_n w_n^{t-1}} \quad \text{with} \quad \alpha^{t-1} = \sqrt{\log \frac{1-\epsilon_n}{\epsilon_n}} \tag{9}$$

A stronger weight is assigned patches with a more stable performance, and the corresponding positions are used to find the gravity center of the whole tracked object. In fact, after we calculate the weight for each local patch, the validity of each patch at time t is based on a decision stump classifier:

$$v_n^t = \left[w_n^t \cdot [\mathcal{D}(\mathbf{z}_n^t) \wedge \mathcal{B}(\mathbf{z}_n^t)] \right] \geq 1/2N \tag{10}$$

The rationale for Eq.10 is that local patches with smaller weights may be noisy for the gravity center calculation, and so they are neglected from the gravity center calculation as illustrated by Eq.1.
In the time t, we also check the local movement based on the calculated gravity center, if it is not consistent with the GMM model, we also delete the patch with the smallest weight and re-calculate again.

4. Experimental results

To demonstrate the effectiveness of the proposed approach, we perform extensive experiments on six sequences from two public datasets. We especially focused on the complex problem of pedestrian motion tracking, although the method we proposed can be applied to any object. In the supplemental material we show videos of the tracking results for the different approaches.

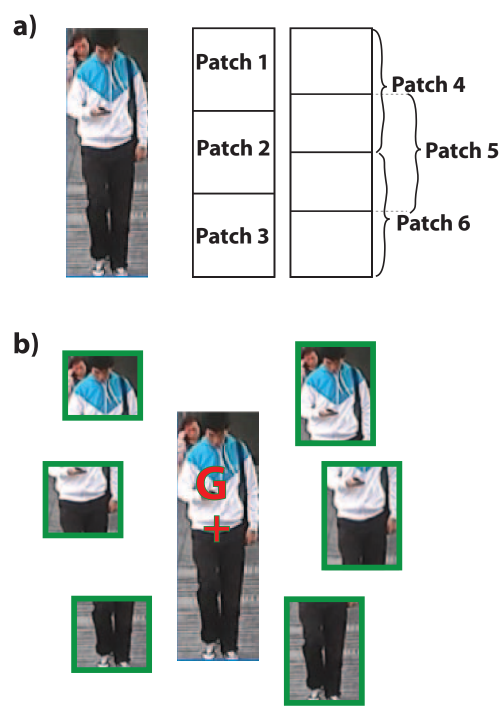

Figure 4. The configuration of the patches z_n ($n = 1 \dots 6$) for an exemplar target.

Initialization and set-up: To avoid the arbitrary influence from using an object detector for initialization, we manually initialized the target objects, however automatic algorithms can be used in practical applications. We initial-

ized the Gaussian mixture model[1] using the displacements $\ell_n^{t=1-15}$ of the first 15 frames and we set K=3 (three components), this value loosely depends on the object and the number of patches. Note that the global tracker can be stable in the initial part of the process, but that's a common assumption of most of the tracking algorithms.
We extracted six overlapping local patches $N = 6$, rigidly dividing bounding box of the targets as illustrated by Fig. 4a-b. Different selections are of course possible however we did not observed much variations, as long as the patches are sufficiently big (at least one fourth of the target window). We described each patch with a stable texture feature information; again, our hypothesis is that by using the combination of several patches, we are able to capture the structure information of the tracked object.

Dataset and evaluation protocol: We compare our algorithm with several base trackers such as the Compressive Tracking [17], Fragtrack [4], Particle filter [18], tracking-learning-detection (TLD) [9] and the STC tracker [19] because they represents the state-of-the-art in the field and because their source codes are freely available from the authors' websites.
Regarding evaluation, image sequences are selected from two public datasets, PETS09 and Robust Fragments. The selected video sequences consists of a variety of challenging cases in complex environments, i.e., occlusions among objects, scale and orientation variations, lighting changes, and clutter given by the background. The first frame of each sequence is shown in Fig.5a. To quantitatively evaluate the performance, we employed tje Displacement Error Rate (DER) for the central location between the tracking result and the ground-truth, in formulae:

$$\mathrm{DER} = \frac{d(O_{x,y}^t, \mathrm{GT}_{x,y}^t)}{\sqrt{A_{\mathrm{GT}}}} = \frac{\sqrt{(x_0 - x_{\mathrm{GT}})^2 + (y_0 - y_{\mathrm{GT}})^2}}{\sqrt{A_{\mathrm{GT}}}},$$

where $O_{x,y}^t$ is the tracking result, $\mathrm{GT}_{x,y}^t$ is the corresponding ground-truth, and A_{GT} represents the area of $\mathrm{GT}_{x,y}^t$ calculated by the width and height of the bounding box (same for all the methods). The lower the DER is, the more accurate the tracking performance is.

ALMM performance: We performed the evaluation on six test sequences characterized by occlusion, quick motion and texture variations. Occlusion is the key issue for the inaccurate localization and instability of tracking algorithms. Another problem is that the textural information may not be powerful enough and here we test whether the local movement information is effective for the detection of the outliers caused by the local tracker.

[1]We used the standard OpenCV implementation

Kalman filter and particle filter are two well-acknowledged tracking algorithms which can handle to some extent partial occlusions, being the latter the be more powerful one. Therefore, in our evaluation, we compare the proposed method with the particle filter based tracking algorithm. As base trackers for our frameworks we exploited STC tracker and Compressive Tracker. Finally, to present a more comprehensive experimental section, we also added the comparisons with TLD and Fragtrack as they are well known tracking algorithms.
Results are reported in Fig.6 and Tab.1; our goal here is to show how ALMMM helps to solve tracking divergence, therefore results must be appreciated in that sense.
In the sequences selected from PETS09 set, `basketball` and `man A-B`, we have to follow the target in a crowded environment (check Fig. 5 for a sample of key-frames). For this reason, during the tracking process, the object is subject to severe image quality deterioration. These sequences are also quite challenging because the background in the scene provides clutter and many objects are similar to the target in appearance. Moreover, severe occlusions occur to the target, which results in the degeneration of its appearance representation in pixel-level space. However, the proposed tracker can handle this issue robustly. From the sequences as shown in Fig. 6, we can conclude that the proposed tracker is effective for visual object tracking and outperforms or equal the state-of-the-art results. Both local movement modelling based methods can achieve better performances than other base trackers, which demonstrates that the proposed approach is more robust to the appearance variation and able to discriminate the target during the tracking.
In the sequence dubbed `red woman`, both ALMMs achieve much better performance than the comparative methods especially at the last frames of the video sequence. The particle filter method also achieves a similar performance as the TLD tracker (sometimes) and compressive tracker. Notice that for this sequence, there is a strong occluding effect caused by crowd. In such regard, local patch based methods can solve this kind of problem in a certain extent.
Finally, from the analysis of the last 2 sequences, ALMM results also robust to textural changes `white woman`, fast motion `basketball` and camera deterioration `woman` (in the last two cases we also have occlusions). Figure 5b reports examples the aforementioned problematic scenarios (occlusions etc) that tracking algorithms may not be able to deal with. As final consideration, the TLD tracker has the strength to relocate on the tracked object when it loses it occasionally. This behavior is fit in the case of a long-time tracking task and it might be included in our approach as future work. Our method achieves in general a high performance, which further demonstrates that the proposed

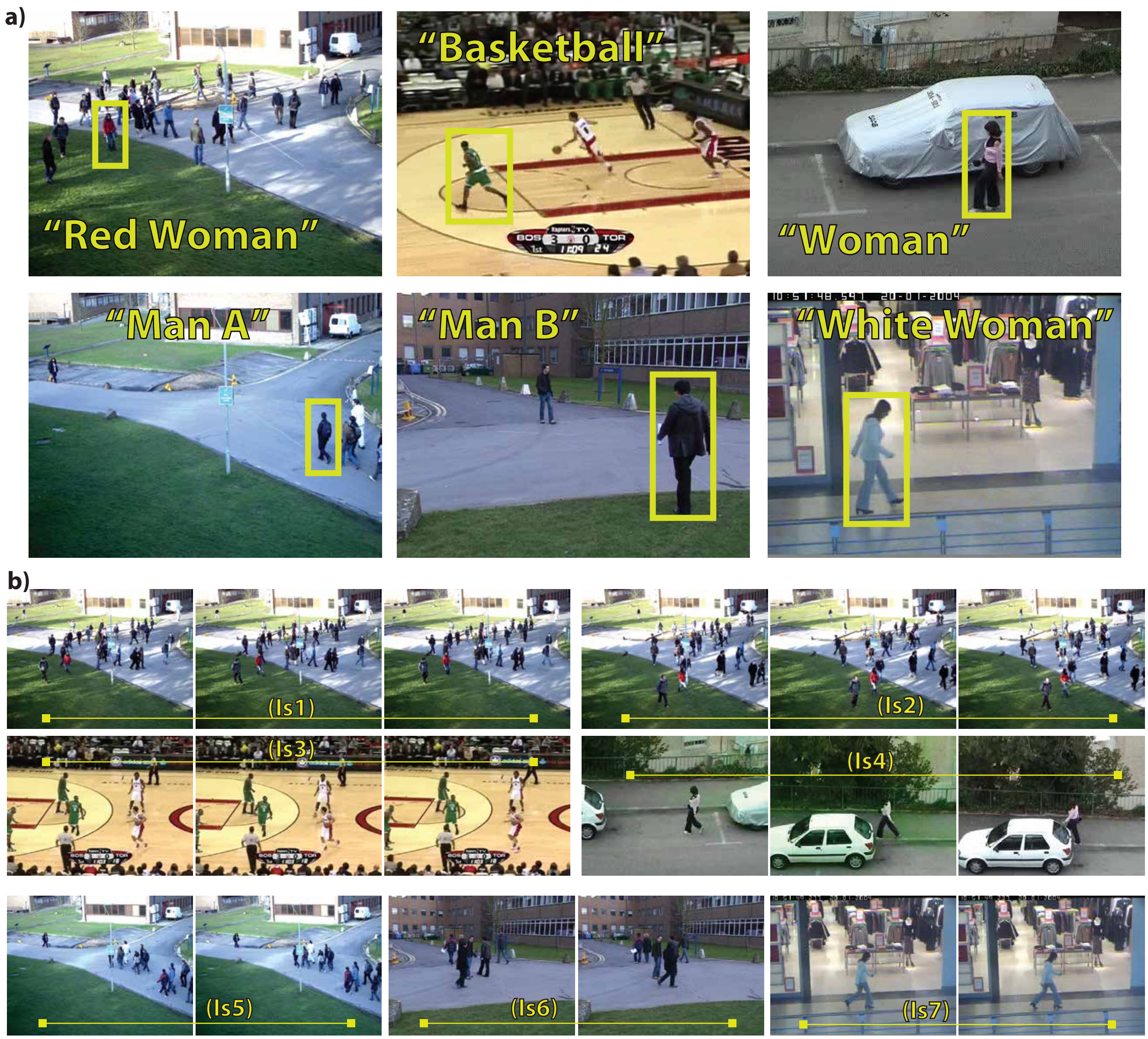

Figure 5. a) The first frame and the tracking target for each sequence considered. b) Key frames where one of the tracking algorithm fails. As visibile failing are mostly due to occlusions or textural issues.

method is more robust to the appearance variation even with severe texture variations and capable to discriminate the target stably during the tracking.
In Table 1, we reported the average and variance of DER of the test data are also used as the measurements to compare the performance of the ALMMs with other trackers. The results confirm that local movement modelling can improve the performance of the tracking system. For example, ALMM with STC tracker achieves a better performance than others, i.e. the variance of ALMM (with STC) is much smaller than that of the base STC tracker on the red woman sequence.

5. Conclusions and future work

This paper presented ALMM, a new framework for object tracking based on adaptive local movement modelling of the local patch. Patch based methods are supposed to be robust to occlusion, but sometime they are not powerful enough since the appearance only information might not be reliable. Our idea was to exploit a local movement model to increase the performance of the patch based tracking

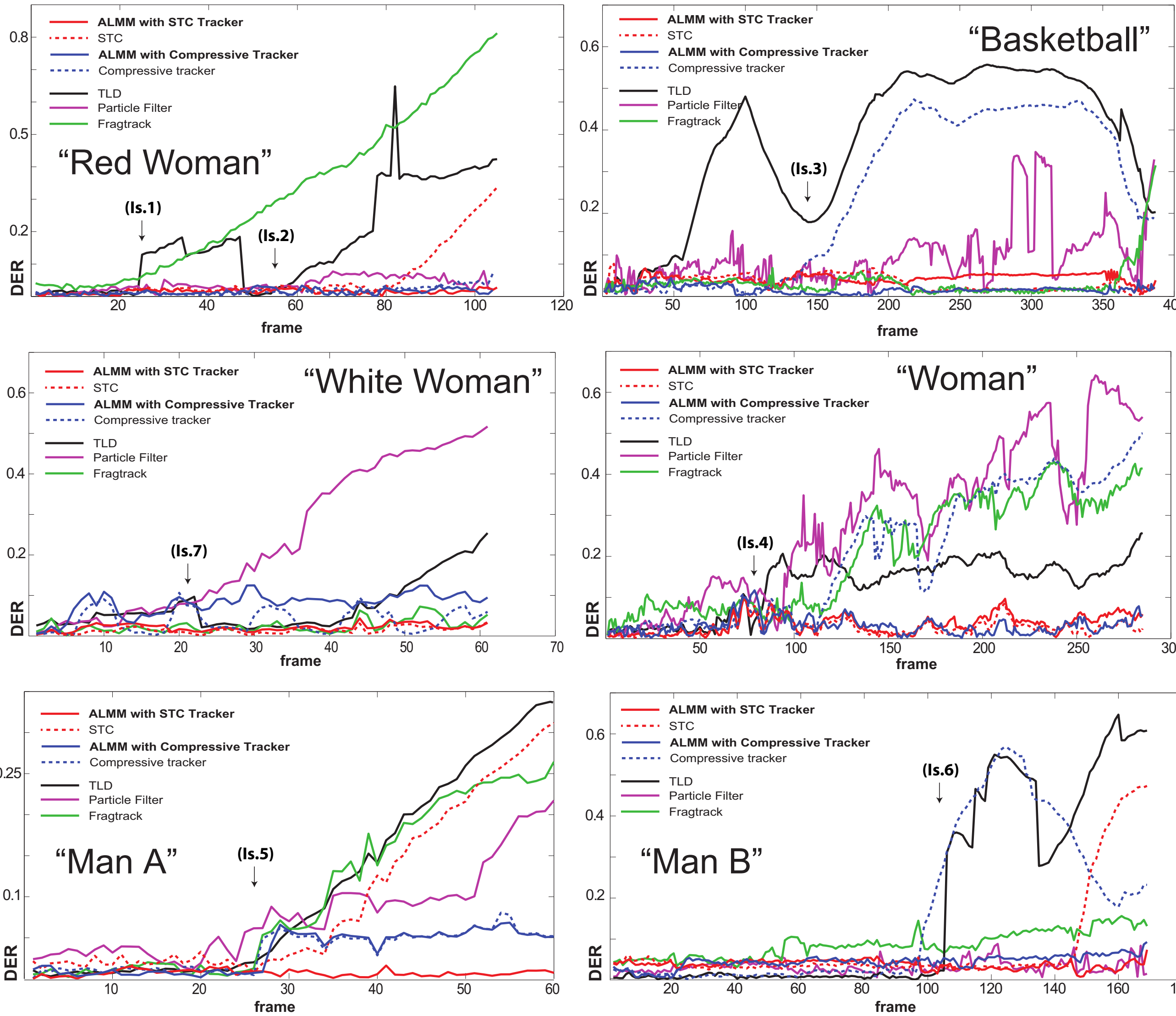

Figure 6. Results for the tracking accuracy in the different tested sequences for our ALMM algorithm, Compressive Tracking [17], Fragtrack [4], Particle filter [18], tracking-learning-detection (TLD) [9] and the STC tracker [19]. The x-axis indicates the number of frames in the sequence while the y-axis refers to the frame number in the sequence. The frames marked with Is.N (issue) are reported in Fig.5b where one can appreciate one a tracking algorithm may diverge

system. In this way a tracking system based on two elements, texture and local movement modelling, embedded into an integrated classifier is verified to be effective in the tracking task. ALMM can be used in conjunction with any part-based tracker, in our experiments we evaluated it with STC and the Compressive trackers. Experimental results unequivocally demonstrate how ALMM alleviated several problems in critical tracking situations, like occlusions. Regarding future work, we will evaluate the proposed method on more datasets with worse image quality and we will develop novel strategies in order to deal with longer video sequences.

Acknowledge

Baochang Zhang thanks for the support by Natural Science Foundation of China, under Contracts 61272052 and 61473086, and by the Program for New Century Excellent Talents University of Ministry of Education of China.

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey ACM Computer Survey, 2006

[2] R. Yao, Q. Shi, C. Shen, Y. Zhang, A. v. hengel Part-based visual tracking with online latent structural learning In Proc. of IEEE CVPR, pp:25-27, 2013

	Red women		Man A	
	Mean $\times 10^{-2}$	Var. $\times 10^{-4}$	Mean $\times 10^{-2}$	Var. $\times 10^{-4}$
ALMM STC	1.48	0.49	0.71	0.15
STC Tracker	5.61	64	9.56	99
ALMM CT	1.41	0.70	3.29	5.76
Compr. Tracker	1.74	1.50	3.48	5.23
TLD	1.65	230	10.12	121
Fragtrack	3.08	586	10.13	93
Part. Filter	3.69	4.49	7.85	29
	White Woman		**Man B**	
	Mean $\times 10^{-2}$	Var. $\times 10^{-4}$	Mean $\times 10^{-2}$	Var. $\times 10^{-4}$
ALMM STC	2.1	0.77	3.81	1.06
STC Tracker	1.44	0.65	7.99	141
ALMM CT	4.54	2.33	1.68	0.76
Compr. Tracker	3.81	8.61	16.03	356
TLD	7.28	38	18.35	547
Fragtrack	2.44	2.175	8.68	10
Part. Filter	23.15	324	3.24	2.42
	Basketball		**Woman**	
	Mean $\times 10^{-2}$	Var. $\times 10^{-4}$	Mean $\times 10^{-2}$	Var. $\times 10^{-4}$
ALMM STC	4.25	1.38	3.45	5.12
STC Tracker	3.40	2.99	2.67	3.71
ALMM CT	1.68	0.76	3.11	4.55
Compr. Tracker	23.9	367	20.9	255
TLD	3.655	307	12.96	46
Fragtrack	3.35	16	20.99	181
Part. Filter	9.5	57	28.48	346

Table 1. Mean and variance of DERs of the comparative methods.

[3] Z.Han, J.Jiao, B.Zhang, Q.Ye and J.Liu Visual object tracking via sample-based adaptive sparse representation Pattern Recognition Journal, 2011

[4] A. Adam, E. Rivlin, and I. Shimshoni Robust fragments-based tracking using the integral histogram In Proc. of IEEECVPR, vol. 1, pp:798-805, 2006.

[5] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang Incremental learning for robust visual tracking International Journal Computer Vision, 77, pp: 125-141, 2008.

[6] J. Kwon and K. M. Lee Visual tracking decomposition In Proc. of IEEE CVPR, pp.1269-1276, 2010.

[7] H. Grabner, M. Grabner, and H. Bischof Semi-supervised online boosting for robust tracking In Proc. of ECCV, pp: 234-247, 2008

[8] X. Wang, G. Hua, and T. X. Han Discriminative tracking by metric learning In Proc. of ECCV, pp: 200-214, 2010.

[9] Z. Kalal, K.Mikolajczyk, and J. Matas Tracking-learning-detection IEEE Trans. PAMI, vol. 34(7), pp:1409-1422, 2012.

[10] D. Sugimura, K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto Using Individuality to Track Individuals: Clustering Individual Trajectories in Crowds Using Local Appearance and Frequency Trait In proc. of IEEE ICCV, 2009.

[11] M. Betke, D. Hirsh, a. Bagchi, N. Hristov, N. Makris,and T. Kunz Tracking Large Variable Numbers of Objects in Clutter In proc. of IEEE CVPR, pp: 1-8, 2007.

[12] S. Ali and M. Shah Floor Fields for Tracking in High Density Crowd Scenes. In proc. of ECCV, 2008.

[13] Y. Rosenberg and M. Werman Representing local motion as a probability distribution matrix for object tracking and other applications In proc. of IEEE CVPR , pp: 654-659, 1997

[14] N. Jacobs, M. Dixon, and R. Pless Location-specific Transition Distributions for Tracking In IEEE Workshop on Motion and Video Computing, 2008.

[15] O. Nestares and D. J. Fleet Probabilistic Tracking of Motion Boundaries With Spatiotemporal Predictions In proc. of IEEE CVPR, pp: 358-365, 2001.

[16] Varun Ramakrishna, Yaser Sheikh, Takeo Kanade Tracking Human Pose by Tracking Symmetric Parts in proc. of IEEE CVPR, 2013.

[17] K. Zhang, L. Zhang and M.H. Yang, Real-time Compressive Tracking, IEEE Trans. PAMI, 36(10),pp.2002-2015,2014.

[18] Gustafson F, Gunnison F, Nicolas B, et al. Particle filters for position, navigation, and tracking IEEE Transactions on Signal Processing, vol. 50 (2), pp:425 -437, 2002.

[19] Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang, and David Zhang, Fast Tracking via Spatio-Temporal Context Learning, in Proc. of ECCV, pp. 127-141, 2014

[20] C. Stauffer and W.E.L. Grimson Adaptive Background Mixture Models for Real-Time Tracking In Proc. of IEEE CVPR, vol. 2, pp. 246-252, 1999

[21] Weijun Wang and Ramakant Nevatia Robust object tracking using constellation model with superpixel In Proc. of ACCV, 2012

[22] B. Zhang, Y. Gao, S. Zhao, B. Zhong Kernel Similarity Modeling of Texture Pattern Flow for Motion Detection in Complex Background IEEE Trans. CSVT, 21(1), pp: 29-38, 2011