Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A robust object tracking framework based on a reliable point assignment algorithm*#

Rong-feng ZHANG[1,2], Ting DENG[†‡3], Gui-hong WANG[1], Jing-lun SHI[1], Quan-sheng GUAN[1]

([1]*School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China*)

([2]*School of Electronic and Information Engineering, Guangzhou College of South China University of Technology, Guangzhou 510800, China*)

([3]*Information Network Engineering and Research Center, South China University of Technology, Guangzhou 510641, China*)

[†]E-mail: dengting@scut.edu.cn

Received Aug. 14, 2016; Revision accepted Dec. 21, 2016; Crosschecked Mar. 28, 2017

Abstract: Visual tracking, which has been widely used in many vision fields, has been one of the most active research topics in computer vision in recent years. However, there are still challenges in visual tracking, such as illumination change, object occlusion, and appearance deformation. To overcome these difficulties, a reliable point assignment (RPA) algorithm based on wavelet transform is proposed. The reliable points are obtained by searching the location that holds local maximal wavelet coefficients. Since the local maximal wavelet coefficients indicate high variation in the image, the reliable points are robust against image noise, illumination change, and appearance deformation. Moreover, a Kalman filter is applied to the detection step to speed up the detection processing and reduce false detection. Finally, the proposed RPA is integrated into the tracking-learning-detection (TLD) framework with the Kalman filter, which not only improves the tracking precision, but also reduces the false detections. Experimental results showed that the new framework outperforms TLD and kernelized correlation filters with respect to precision, f-measure, and average overlap in percent.

Key words: Local maximal wavelet coefficients; Reliable point assignment; Object tracking; Tracking learning detection (TLD); Kalman filter

http://dx.doi.org/10.1631/FITEE.1601464 **CLC number:** TP391.41

1 Introduction

Visual tracking is the locating and tracing of one or multiple objects in video sequences. In recent years, with the development of computer hardware and image processing, visual tracking has become one of the most active research areas in computer vision. It has been widely applied in many fields, e.g., intelligence surveillance (Jeong *et al.*, 2014; Prakash and Thamaraiselvi, 2014), traffic monitoring (Kaur and Sahambi, 2015), public security (Xu and Gao, 2010), anti-terrorism (Elhamod and Levine, 2013; Jung and Yoon, 2015), advanced driver assistance systems (ADASs) (Elmenreich and Koplin, 2011), and human computer interaction (Cheng *et al.*, 2016). However, tracking objects is a complicated task, especially for long-term tracking. The technology still faces various challenges, such as illumination change, object occlusion, appearance deformation, scale change, background clutter, camera viewpoint change, and real-time processing.

Numerous approaches for visual tracking have been proposed in the literature over the past two decades, and great achievements have been made. Some of these approaches, such as the optical flow

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61671213 and 61302058) and the Guangzhou Key Lab of Body Data Science (No. 201605030011)

A preliminary version was presented at the 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, Oct. 15–17, 2016, Datong, China

ORCID: Ting DENG, http://orcid.org/0000-0001-9394-5430

tracker (Brox *et al.*, 2004), mean shift tracker (Yu *et al.*, 2015), tracking-learning-detection (TLD) tracker (Kalal *et al.*, 2012), and kernelized correlation filters (KCF) tracker (Henriques *et al.*, 2015), have performed well in solving the tracking problems. The optical flow tracker does well in tracking a moving object against an unaware background, but it is sensitive to illumination change and has an expensive time calculation. The mean shift tracker performs well in real-time processing and partial occlusion, but it will fail when the scale changes or the object is fully occluded and then reappears. The KCF tracker performs well in real-time processing, illumination change, image noise, and background clutter, but the performance during scale change and appearance deformation is not good. KCF may fail in object tracking if the target disappears from the video and reappears. Recently, the proposed TLD method has become a popular visual tracking algorithm because it has been shown to provide promising performances in object occlusion, scale change, and background clutter. However, experiments show that this method is sensitive to changing illumination (Jia *et al.*, 2015). Moreover, the high computational cost of the algorithm prevents it from being used at higher resolutions and frame rates. Recently, a reliable part-based tracking strategy (Li *et al.*, 2015; Liu *et al.*, 2015) was proposed to track objects by modeling the target's appearance based on multiple parts. This strategy successfully resolves the tracking problem of partial occlusions. However, as Liu *et al.* (2016) pointed out, the performance of this strategy depends on the tracker being employed. Tracking objects by machine learning has been another research topic over these years. Tarkov and Dubynin (2013) proposed an algorithm for tracking objects in real time based on back propagation neural networks and used graphics processing unit (GPU) for speed acceleration. Experimental results show that this approach can successfully track objects in real time. However, the performance of this method is strongly dependent on the training dataset and GPU hardware. More recently, Redmon *et al.* (2016) and Ning *et al.* (2016) presented deep learning based approaches for object detection and tracking. Since these methods detect or track the object using deep learning approaches and use GPU to increase the speed, their performance is notable. On the other hand, the performance of these methods strongly depends on the training dataset and the hardware. Furthermore, as the authors claimed, these methods had limitations in detecting small objects that appear in groups, and they can struggle to generalize objects in new or unusual aspect ratios or configurations.

Because an increasing number of applications are going to employ visual tracking in the future, a number of issues remain to be solved. The aim of this study is to improve the performance of object tracking in cases of illumination change, object occlusion, appearance deformation, and real-time processing with an improved TLD framework. To do this, a reliable point assignment (RPA) algorithm based on a wavelet transform is proposed to improve the performance of the tracker. By taking advantage of wavelet transform properties, the proposed RPA decomposes the input image region, which includes the tracking target, by several scales. The reliable points are located by searching the points that possess local maximal wavelet coefficients. Due to the fact that the wavelet coefficient denotes image variation, the reliable points are rich with discriminant information for tracking, which makes the points robust against image noise, illumination change, and change in target size. Hence, as shown in our previous work (Zhang *et al*., 2016), RPA improves the tracking precision. However, the experimental results in Zhang *et al*. (2016) showed that the false detection of RPA needs to be further improved and the processing needs to be speeded up. Therefore, to reduce the processing time and false detections, a Kalman filter is applied to the tracking and detection model, which can greatly reduce the search space of the detection model. By cutting down the search space, false detection results from the detector in TLD decrease because background interference is reduced. Owing to RPA and the fusion of the Kalman filter, the learning step is improved. Consequently, the method proposed in this study is faster and more robust against image noise, illumination change, and tracking target size than the original TLD.

The main contributions of this paper are as follows:

1. A wavelet transform based RPA algorithm is proposed for the tracker, which makes the TLD framework robust against image noise, illumination change, and tracking target size.

2. The Kalman filter is adopted to fuse with the tracking and detection model, which not only speeds up the processing of TLD framework, but also improves the detection performance.

2 Related works

The TLD algorithm was proposed for long-term tracking of a single target. Fig. 1 illustrates its main procedure. One of the advantages of the system is that it does not need to separate an offline learning stage. As shown in Fig. 1, there are mainly three steps in the TLD algorithm: tracking, detection, and feedback learning. To initialize tracking, a bounding box is drawn to include the target object and an equally spaced set of points (10×10) is constructed in the bounding box for tracking. The tracking component based on the Median Flow tracker (Kalal *et al.*, 2010a; Yu and Zeng, 2015) is used to obtain a tracking bounding box that covers the target object.

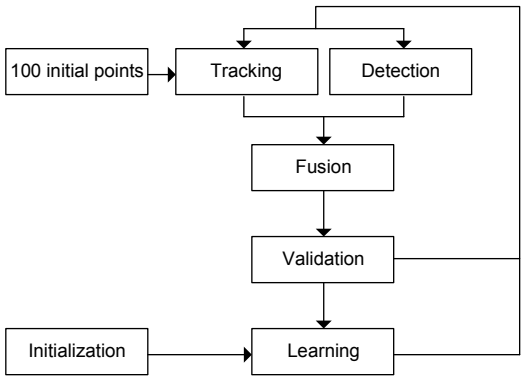

Fig. 1 Procedure for the tracking-learning-detection (TLD) framework

The detecting component is based on a sliding-window approach (Viola and Jones, 2001; Dalal and Triggs, 2005) that can generate up to 200 000 sub-windows for a 640×480 video graphics array (VGA) image resolution. The sub-windows are tested by a cascade procedure that includes a variance filter, a random forest classifier, and a normalized cross correlation (NCC) filter to produce one or more detecting bounding boxes. If the tracker tracks the target successfully, the detecting bounding boxes will fuse with the tracking bounding box to generate a final result. Otherwise, the detector is considered to be successful only if it generates only one detecting bounding box. In this way, if the object disappears from the camera view, the detector will capture the object and recover the tracking process when the target reappears.

For long-term tracking, the target object may change its appearance, which can result in tracking and detection failures. Therefore, TLD introduces an online learning approach (P-N learning) that uses the tracker and detector results to generate positive (P) and negative (N) examples, which are added to the model of the detector to produce stable outputs.

3 The proposed method

The TLD algorithm has demonstrated good performance in long-term object tracking. However, as described in Section 2, 100 points having equal space are constructed in the initial step. This may be sensitive to image noise and illumination change since the position of the point may be coincidently located on an image noise or an image area that is sensitive to illumination. Therefore, a more robust point assignment algorithm is drastically needed so as to improve the performance of the tracking step. Another limitation of TLD is that the detector needs to verify up to 200 000 sub-windows for searching the target, which is a bottleneck in real-time applications. To overcome these two limitations, we propose a wavelet-based reliable point assignment approach so that the positions of the initial points can be robust against image noise and illumination change. Moreover, to reduce the tracking time and false detections, a Kalman filter is used to cooperate with the tracker and detector.

3.1 Reliable point assignment

Corner or blob detectors such as the Harris corner detector (Harris and Stephens, 1988), Laplacian of Gaussian (LoG) detector (Kong *et al.*, 2013), difference of Gaussian (DoG) detector (Lowe, 2004), and speeded-up robust features (SURF) detector (Bay *et al.*, 2008) are used in the first step for many tracking algorithms. Most of the detectors are based on the second derivatives of the image, and for multiscale analysis, the Gaussian function (or approximation of the Gaussian, such as SURF) is applied to construct an image pyramid. Due to the complex computation

of these detectors, they are limited to real-time tracking. Moreover, since these detectors try to search interest points based on the corner or blob regions, their performances in tracking are not reliable due to the fact that not all the tracking objects have corners or blobs inside. Furthermore, the size of the tracking object may change severely, and as a result, the number of interest points can vary drastically. Specifically, for those very small tracking objects, there can be very few points generated by these detectors, which leads to tracking failures.

To determine interest points that have distinguished information and whose locations are reliable when the object lacks corners or blobs, or the object is of small size, an ingenious point location algorithm based on wavelet transform is proposed in this study.

The wavelet transform is a multi-resolution representation that can be applied to study the signal at different scales using scaling functions and wavelet functions. Suppose that $f(x)$ is a discrete signal. The discrete wavelet transform (DWT) coefficients of $f(x)$ are defined as (Details can be found in Gonzalez and Woods (2002))

$$W_\psi(j,k)=h_\psi(-n)*W_\varphi(j+1,n)\big|_{n=2k,k\geq 0}, \quad (1)$$

$$W_\varphi(j,k)=h_\varphi(-n)*W_\varphi(j+1,n)\big|_{n=2k,k\geq 0}, \quad (2)$$

where $W_\varphi(j, k)$ is the scaling coefficient (approximation of $f(x)$) obtained by convoluting $f(x)$ with low pass filter $h_\varphi(-n)$ at scale j and down sampling the result by 2. $W_\psi(j, k)$ is the wavelet coefficient (details for $f(x)$) obtained by convoluting the function with high pass filter $h_\psi(-n)$ at scale j and down sampling the result by 2. Fig. 2 shows a DWT procedure, which applies Eqs. (1) and (2) iteratively.

To obtain the wavelet coefficients of an image, a one-dimensional transform is performed on the rows following on the columns. As a result, an input image can be transformed into an approximation image, and three detail images corresponding to the image variation in horizontal, vertical, and diagonal directions. By transforming the image at different scales, we can study the image at different resolutions with the approximation and detail information.

The intention of RPA is to find the reliable points in the image that have as much distinguished information as possible using wavelet transformation.

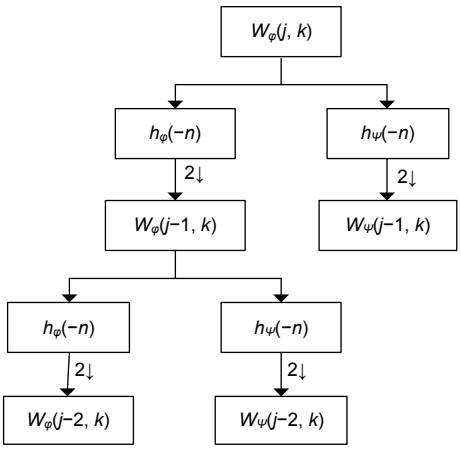

Fig. 2 A discrete wavelet transform procedure

As described above, the detailed images at different scales represent the image variation at different resolutions, and a high wavelet coefficient (in absolute value) at a low resolution is related to a region with high global variation. Therefore, it is obvious that any point in the image related to high global variation is robust to image noise and illumination change. On the other hand, a high wavelet coefficient (in absolute value) at a high resolution provides detailed information of the image, which can be used to discriminate a local region from another. Thus, by tracing a point's wavelet coefficients from a high resolution to a low resolution, the point is more discriminative and robust if the summation of its wavelet coefficients at different scales is greater than the other points. Consequently, a straightforward approach is to decompose the input image at several scales and find those points that have a high summation of wavelet coefficients by thresholding. However, this is not correct for the object tracking case because the number of points may vary seriously with respect to object appearance, and the location of the points may converge to the strong edge. Fig. 3a shows a sub-region (marked by a white rectangle) for tracking in a video frame and Fig. 3b the interest points for tracking using the TLD approach. Fig. 3c is the interest point location by using the top-100 high-wavelet-coefficient points. As we can see, though, the points in Fig. 3c are those with high local variation, and the points converge to limited areas. This is not a good selection since some information from the sub-region is discarded.

Therefore, we propose to find the reliable points based on the local maximal wavelet coefficients. To do this, the input image (or an image region) is separated into some non-overlapping regions with equal sizes. For a point in a sub-region, the wavelet coefficients are summed up from the highest resolution to the lowest resolution (Fig. 4).

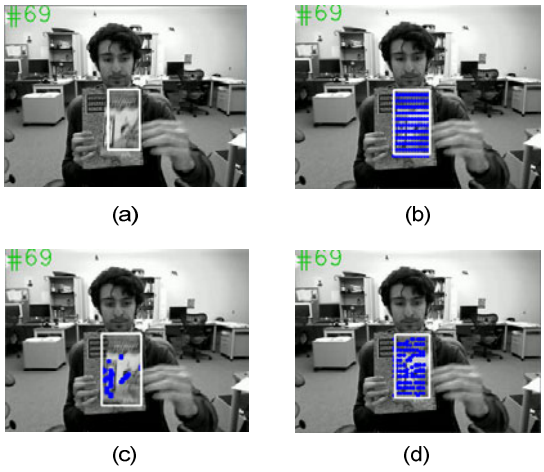

Fig. 3 Interest point assignment: (a) sub-region for tracing; (b) TLD interest points; (c) interest points for the highest wavelet coefficients; (d) interest points for the proposed method

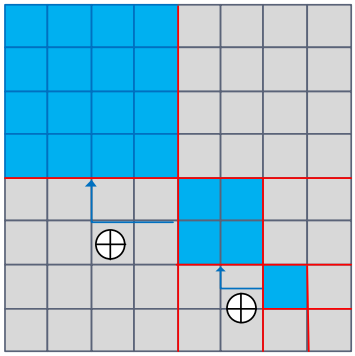

Fig. 4 Summation of wavelet coefficients for a point at each scale

Suppose that an input image is decomposed by n scales, and for a point $p(x, y)$ in the image, the wavelet coefficient set $S(x, y)$ of $p(x, y)$ is defined as

$$S(x,y)=\left\{W_{\mathrm{h}}(x*2^{j},y*2^{j}),W_{\mathrm{v}}(x*2^{j},y*2^{j}),\right.\\ \left.W_{\mathrm{d}}(x*2^{j},y*2^{j})\right\}, \tag{3}$$

where j ($-n\leq j\leq -1$) denotes the wavelet scale and $W_{\mathrm{h}}(x*2^{j}, y*2^{j})$, $W_{\mathrm{v}}(x*2^{j}, y*2^{j})$, and $W_{\mathrm{d}}(x*2^{j}, y*2^{j})$ are the details in horizontal, vertical, and diagonal directions, respectively. Therefore, the summation of the details for $p(x, y)$ can be defined as

$$\mathrm{Sum}(x,y)=\sum_{j=-n}^{-1}\left[W_{\mathrm{h}}(x*2^{j},y*2^{j})+W_{\mathrm{v}}(x*2^{j},y*2^{j})\right.\\ \left.+W_{\mathrm{d}}(x*2^{j},y*2^{j})\right]. \tag{4}$$

Suppose that (l, t) is the left-top coordination of a sub-region $R(x, y)$ in the image, and that the width and height of $R(x, y)$ are r_{w} and r_{h}, respectively. The local maximum is then defined as

$$E_{R(x,y)}=\max\{\mathrm{Sum}(x,y)\}, \tag{5}$$

where $l\leq x<l+r_{\mathrm{w}}$, $t\leq y<t+r_{\mathrm{h}}$.

As described in Eq. (5), the point corresponding to the local maximum is defined as the reliable point in this paper. Since the reliable point is located by searching the local maximal wavelet coefficient in each non-overlapping sub-region, the number of points for object tracking can be pre-determined and it is stable during the tracking. Furthermore, due to the selection of the maximal summation of all the details in all the scales, the point clearly has rich information for tracking, which leads to an improvement in tracking performance. For real-time processing, the Haar wavelet is chosen in this study for simplicity. Fig. 3d is an example of the reliable point for the sub-region of Fig. 3a.

3.2 Integration of the Kalman filter and RPA into TLD

The Kalman filter is an estimator that provides an efficient recursive method to estimate the state of a linear process, in a way that minimizes the mean of the squared error (Kalman, 1960). The Kalman filter is typically divided into two stages. One is time update (prediction), and the other is measurement update (correction). Time update advances the state based on the state equation until the next measurement is obtained. Measurement update incorporates the measurement from sensors based on the measurement equation (Jeong *et al.*, 2014). It has been widely used to estimate the position of an object in each frame of the sequence (Sun *et al.*, 2010).

As mentioned in Section 2, the detection step in TLD needs to perform a variance filter, a random forest classifier, and an NCC filter for many sub-windows, which is a bottleneck in real-time tracking. Therefore, to speed up the detection step and improve the tracking performance, in this study we integrate the Kalman filter and RPA into the TLD framework. The proposed framework is illustrated in Fig. 5.

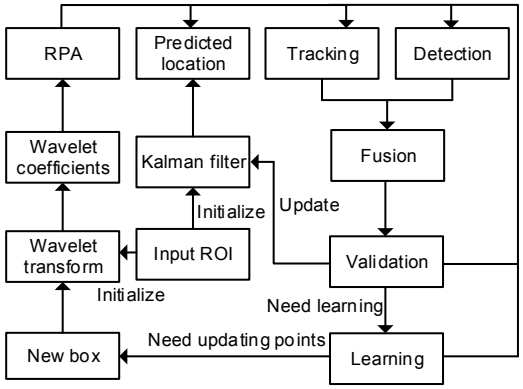

Fig. 5 The proposed tracking framework

For an input region of interest (ROI), the reliable points are assigned by the proposed RPA based on the Haar wavelet transform and the location of this ROI in the frame is used to initialize the Kalman filter first. During the object tracking procedure, the tracking step described in Section 2 is first applied and the Kalman filter is updated when tracking is successful (the Kalman filter is set to invalid when tracking has failed). Next, the location predicted by the Kalman filter is used in the detection step if the Kalman filter is valid for the current frame. Instead of detecting all sub-windows of the current frame, in this study we check only those sub-windows that overlap with the predicted location using the variance filter, random forest classifier, and NCC filter. If there are still valid sub-windows after applying these three filters, the sub-windows are used to fuse with the tracking result. Otherwise, these three filters are applied to other sub-windows that do not overlap with the predict location, which executes in the same way as the original TLD method does. The fusion result of the detection and tracking is then checked and the learning step is applied if it is necessary. During the learning step, if any new appearance of the object is found, the reliable point is then re-assigned, and the positions of the points are updated for the tracking step. With this strategy, the new framework does not need to compute the reliable points frequently. As a result, less time is consumed with RPA.

4 Results

4.1 Dataset

The video sets for performance evaluation include 9 videos (Kalal *et al.*, 2010b) and 50 videos (http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html), respectively, which also provide the ground truth data and the initial tracking box position. Note that since there are four reduplicate videos in both datasets, the number of videos for the experiment is 55. Table 1 shows the information from the testing videos and Fig. 6 shows some example frames from the videos.

4.2 Experimental environment

The code for the proposed method is written in C++. For a fair comparison, TLD and KCF are also tested using C++ code downloaded from https://github.com/arthurv/OpenTLD and https://github.com/victorgan/henriques2014high, respectively. Table 2 shows the experimental environment for the proposed method, TLD, and KCF.

4.3 Evaluation criteria

The performance of the proposed method is evaluated by comparison with those of TLD and KCF. The evaluation criteria include precision (P), recall (R), f-measure (F), frame rate (FR, frame/s), average overlap in percent (AOP), and average number of failures in 1000 frames (ANF) (http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html). The method of calculating AOP is shown in Fig. 7 and P, R, F, and FR are defined in Eqs. (6)–(9):

$$P = \mathrm{TP} / \mathrm{RN}, \tag{6}$$

$$R = \mathrm{TP} / \mathrm{ON}, \tag{7}$$

$$F = 2PR / (P + R), \tag{8}$$

$$\mathrm{FR} = 1000 / F_{\mathrm{t}}, \tag{9}$$

where TP represents the true positives of the method, RN is the number of all responses of the method, ON

Table 1 Information from the testing videos

Video index	Video	Frame number	Number of occurrences*	Resolution
1	Basketball	725	725	576×432
2	Biker	142	142	640×360
3	Bird1	408	408	720×400
4	BlurBody	334	334	640×480
5	BlurCar2	585	585	640×480
6	BlurFace	493	493	640×480
7	BlurOwl	631	631	640×480
8	Bolt	350	350	640×360
9	Box	1161	1161	640×480
10	Car1	1020	1020	320×240
11	Car4	659	659	360×240
12	CarDark	393	393	320×240
13	CarScale	252	252	640×272
14	ClifBar	472	472	320×240
15	Couple	140	140	320×240
16	Crowds	347	347	600×480
17	David	761	761	320×240
18	Deer	71	71	704×400
19	Diving	231	215	400×224
20	DragonBaby	113	113	640×360
21	Dudek	1145	1145	720×480
22	Football	362	362	624×352
23	Freeman4	297	283	360×240
24	Girl	500	500	128×96
25	Human3	1698	1698	480×640
26	Human4	667	667	640×480
27	Human6	792	792	480×640
28	Human9	305	305	320×240
29	Ironman	166	166	720×304
30	Jump	122	122	416×234
31	Jumping	313	313	352×288
32	Liquor	1741	1741	640×480
33	Matrix	100	100	800×336
34	MotorRolling	164	164	640×360
35	Panda	3000	2730	312×233
36	RedTeam	1918	1918	352×240
37	Shaking	365	365	624×352
38	Singer2	366	366	624×352
39	Skating1	400	400	640×360
40	Skating2	473	473	640×352
41	Skiing	81	81	640×360
42	Soccer	392	392	640×360
43	Surfer	376	376	480×360
44	Sylvester	1345	1345	320×240
45	Tiger2	365	365	640×480
46	Trellis	569	569	320×240
47	Walking	412	412	768×576
48	Walking2	500	500	384×288
49	Woman	597	597	352×288
50	Pedestrian2	338	266	320×240
51	Pedestrian3	184	156	320×240
52	Car	945	860	320×240
53	Motorcross	2665	1412	470×310
54	Volkswagen	8576	5141	640×240
55	Carchase	9928	8660	290×217

* Number of frames where the tracking object appears

is the number of object occurrences that should have been detected, and F_t is the average time consumption (in ms) of a frame. Following Kalal *et al.* (2012), a resulting box is considered to be correct if its overlap with the ground truth bounding box is larger than 25% (Fig. 7).

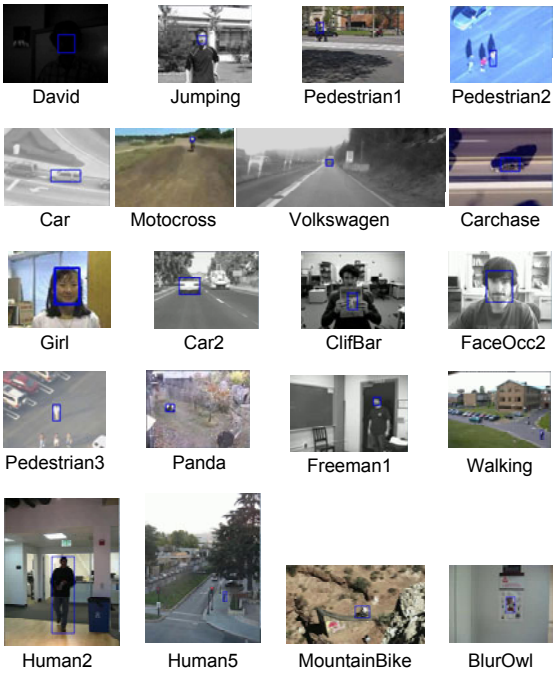

Fig. 6 Some frames from the video data

Table 2 The experimental environment

Item	Description
CPU	Intel® Core™ i7-4710MQ, 2.50 GHz
Memory	16 GB
Hard disk	1000 GB
OS	Windows 8, 64-bit
Programming tool	VS2013, OpenCV 2.4.10

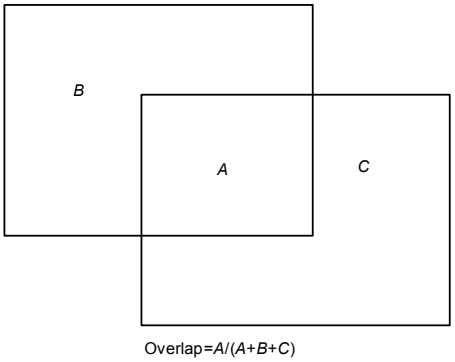

Fig. 7 The overlap of two bounding boxes

4.4 Tracking results

The tracking results for TLD, KCF, and the proposed method are shown in Tables 3–5. Table 6 shows an overall comparison of the proposed method, KCF, and TLD with respect to the precision, recall, f-measure, FR, and AOP. Figs. 8–12 also show the precision, recall, f-measure, FR, and AOP of the three methods.

As we can see from Tables 3 and 6, the average precision of the proposed method is 84.63%, which is 6.54% and 21.83% higher than those of TLD and KCF methods, respectively. This is owing to the use of RPA, which provides reliable points for tracking. Since the tracking performance is improved, the precision is also promoted. For the KCF method, since this method tracks the object using a simple training and detecting framework, it generated more false detection results than the other methods. Furthermore, the simple tracking framework of KCF made the method fail to track the target when it disappeared from the video and reappeared (such as Motocross, Volkswagen, and Carchase videos). Therefore, the precision of KCF is lower than that of the other two methods.

For the recall results, the proposed method outperformed the TLD method by 12.03%, due to the improvement in the tracking process and the use of a Kalman filter to reduce false detections (Table 6). Compared to the KCF method, the proposed method obtained a 3.76% lower recall than KCF did (Table 6). The reason is that the proposed method uses a variance filter, a random forest classifier, and an NCC filter to verify the tracking result (as TLD does), which, on one hand, can help to reduce false detection, and on the other hand, can lead to some rejection of the tracking results. Compared to the TLD method,

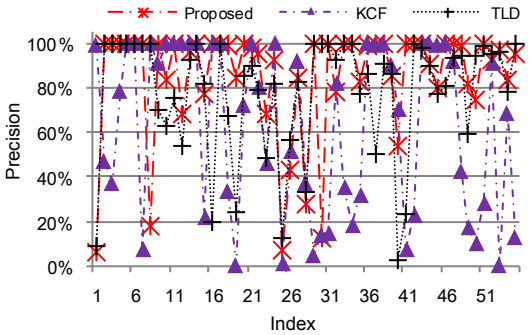

Fig. 8 Precision of TLD, KCF, and the proposed method

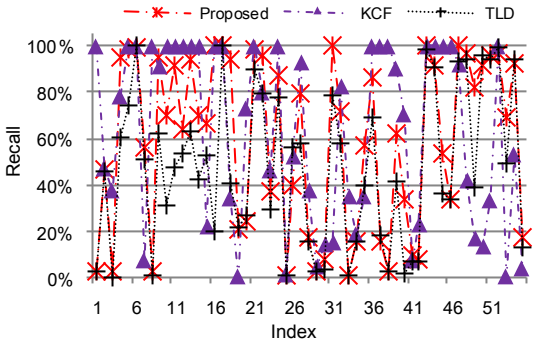

Fig. 9 Recall for TLD, KCF, and the proposed method

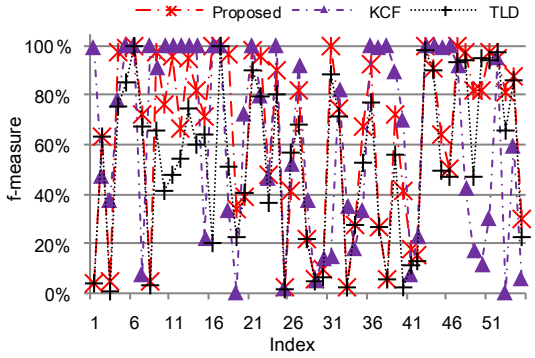

Fig. 10 The f-measure for TLD, KCF, and the proposed method

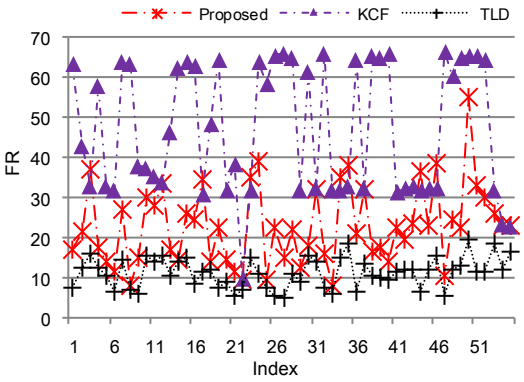

Fig. 11 FR for TLD, KCF, and the proposed method

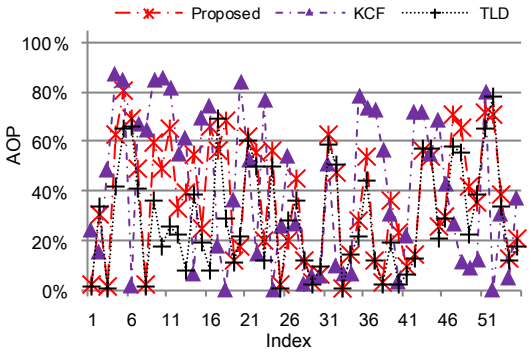

Fig. 12 AOP of TLD, KCF, and the proposed method

Table 3 Tracking results of the proposed method

Video	Response	ON	TP	*P*	*R*	*F*	FR	AOP	ANF
Basketball	329	725	20	6.08%	2.76%	3.80%	17.16	2.07%	892.41
Biker	66	142	66	100.00%	46.48%	63.46%	21.41	30.01%	535.21
Bird1	10	408	10	100.00%	2.45%	4.78%	36.89	1.30%	975.49
BlurBody	318	334	318	100.00%	95.21%	97.55%	17.24	62.94%	47.90
BlurCar2	575	585	574	99.83%	98.12%	98.97%	14.01	80.90%	18.80
BlurFace	490	493	490	100.00%	99.39%	99.69%	11.61	68.97%	6.09
BlurOwl	353	631	353	100.00%	55.94%	71.75%	26.91	48.87%	440.57
Bolt	51	350	9	17.65%	2.57%	4.49%	7.67	2.01%	965.71
Box	1099	1161	1098	99.91%	94.57%	97.17%	14.77	59.77%	121.45
Car1	852	1020	711	83.45%	69.71%	75.96%	30.19	49.26%	266.67
Car4	604	659	604	100.00%	91.65%	95.65%	28.16	65.49%	83.46
CarDark	369	393	252	68.29%	64.12%	66.14%	33.40	33.14%	302.80
CarScale	246	252	236	95.93%	93.65%	94.78%	16.91	38.89%	35.71
ClifBar	329	472	329	100.00%	69.70%	82.15%	14.28	54.39%	302.97
Couple	121	140	93	76.86%	66.43%	71.26%	26.23	24.55%	271.43
Crowds	347	347	346	99.71%	99.71%	99.71%	24.70	66.01%	0.00
David	761	761	761	100.00%	100.00%	100.00%	34.42	56.30%	0.00
Deer	68	71	67	98.53%	94.37%	96.40%	13.96	68.04%	56.34
Diving	53	215	45	84.91%	20.93%	33.58%	22.57	12.08%	753.49
DragonBaby	27	113	27	100.00%	23.89%	38.57%	14.44	17.36%	761.06
Dudek	1145	1145	1127	98.43%	98.43%	98.43%	11.45	62.20%	0.00
Football	362	362	346	95.58%	95.58%	95.58%	9.56	55.57%	0.00
Freeman4	156	283	107	68.59%	37.02%	48.09%	35.27	19.51%	621.91
Girl	472	500	437	92.58%	87.40%	89.92%	39.13	56.43%	56.00
Human3	299	1698	21	7.02%	1.24%	2.10%	9.36	1.70%	918.14
Human4	623	667	266	42.70%	39.88%	41.24%	22.71	19.78%	565.22
Human6	744	792	629	84.54%	79.42%	81.90%	15.13	45.04%	176.77
Human9	192	305	54	28.13%	17.70%	21.73%	21.75	11.43%	763.93
Ironman	5	166	5	100.00%	3.01%	5.85%	12.43	2.79%	969.88
Jump	83	122	10	12.05%	8.20%	9.76%	18.20	6.27%	860.66
Jumping	313	313	313	100.00%	100.00%	100.00%	32.01	62.83%	0.00
Liquor	1578	1741	1240	78.58%	71.22%	74.72%	15.77	47.10%	240.67
Matrix	1	100	1	100.00%	1.00%	1.98%	7.77	0.42%	990.00
MotorRolling	26	164	26	100.00%	15.85%	27.37%	35.06	14.40%	841.46
Panda	1904	2730	1567	82.30%	57.40%	67.63%	37.90	27.75%	349.82
RedTeam	1670	1918	1659	99.34%	86.50%	92.47%	20.73	53.95%	129.30
Shaking	57	365	57	100.00%	15.62%	27.01%	32.26	11.98%	843.84
Singer2	11	366	11	100.00%	3.01%	5.84%	16.45	2.77%	969.95
Skating1	293	400	249	84.98%	62.25%	71.86%	17.38	35.73%	340.00
Skating2	296	473	160	54.05%	33.83%	41.61%	13.87	22.89%	623.68
Skiing	8	81	8	100.00%	9.88%	17.98%	22.73	9.68%	901.23
Soccer	32	392	32	100.00%	8.16%	15.09%	19.59	13.98%	918.37
Surfer	375	376	375	100.00%	99.73%	99.87%	23.24	56.26%	2.66
Sylvester	1345	1345	1225	91.08%	91.08%	91.08%	36.33	53.60%	14.13
Tiger2	244	365	196	80.33%	53.70%	64.37%	22.84	25.78%	339.73
Trellis	190	569	190	100.00%	33.39%	50.07%	38.55	29.33%	666.08
Walking	412	412	411	99.76%	99.76%	99.76%	10.35	70.86%	0.00
Walking2	488	500	481	98.57%	96.20%	97.37%	24.39	66.30%	38.00
Woman	596	597	489	82.05%	81.91%	81.98%	22.45	41.41%	122.28
Pedestrian2	325	266	243	74.77%	91.35%	82.23%	55.02	35.23%	22.56
Pedestrian3	149	156	149	100.00%	95.51%	97.70%	32.85	71.93%	44.87
Car	899	860	833	92.66%	96.86%	94.71%	30.06	71.13%	31.40
Motorcross	1014	1412	981	96.75%	69.48%	80.87%	25.92	38.18%	303.82
Volkswagen	5663	5141	4738	83.67%	92.16%	87.71%	23.16	12.91%	63.80
Carchase	1614	8660	1537	95.23%	17.75%	29.92%	22.86	20.33%	822.06
Average	557.31	818.44	483.31	84.63%	58.97%	63.48%	22.94	41.84%	388.90

Table 4 Tracking results for KCF

Video	Response	ON	TP	*P*	*R*	*F*	FR	AOP	ANF
Basketball	725	725	720	99.31%	99.31%	99.31%	63.44	24.55%	0.00
Biker	142	142	67	47.18%	47.18%	47.18%	42.93	15.14%	521.13
Bird1	408	408	153	37.50%	37.50%	37.50%	32.79	48.77%	375.00
BlurBody	334	334	262	78.44%	78.44%	78.44%	57.70	87.04%	161.68
BlurCar2	585	585	585	100.00%	100.00%	100.00%	32.68	85.02%	0.00
BlurFace	493	493	493	100.00%	100.00%	100.00%	31.73	1.92%	0.00
BlurOwl	631	631	46	7.29%	7.29%	7.29%	63.82	67.14%	925.52
Bolt	350	350	350	100.00%	100.00%	100.00%	63.53	64.90%	0.00
Box	1161	1161	1056	90.96%	90.96%	90.96%	38.00	85.09%	66.32
Car1	1020	1020	1020	100.00%	100.00%	100.00%	37.51	85.61%	0.00
Car4	659	659	659	100.00%	100.00%	100.00%	35.07	82.12%	0.00
CarDark	393	393	393	100.00%	100.00%	100.00%	33.71	55.18%	0.00
CarScale	252	252	252	100.00%	100.00%	100.00%	46.61	61.20%	0.00
ClifBar	472	472	472	100.00%	100.00%	100.00%	62.49	6.75%	0.00
Couple	140	140	31	22.14%	22.14%	22.14%	63.93	69.66%	714.29
Crowds	347	347	347	100.00%	100.00%	100.00%	62.97	74.14%	0.00
David	761	761	761	100.00%	100.00%	100.00%	30.86	18.20%	0.00
Deer	71	71	24	33.80%	33.80%	33.80%	48.43	0.18%	591.55
Diving	215	215	1	0.47%	0.47%	0.47%	64.71	36.76%	995.35
DragonBaby	113	113	82	72.57%	72.57%	72.57%	31.79	84.39%	168.14
Dudek	1145	1145	1145	100.00%	100.00%	100.00%	38.42	52.83%	0.00
Football	362	362	290	80.11%	80.11%	80.11%	9.56	14.47%	74.59
Freeman4	283	283	132	46.64%	46.64%	46.64%	31.93	77.26%	321.55
Girl	500	500	500	100.00%	100.00%	100.00%	63.85	0.26%	0.00
Human3	1698	1698	26	1.53%	1.53%	1.53%	58.46	26.31%	978.21
Human4	667	667	347	52.02%	52.02%	52.02%	65.48	53.92%	470.76
Human6	792	792	732	92.42%	92.42%	92.42%	66.03	26.41%	51.77
Human9	305	305	114	37.38%	37.38%	37.38%	64.81	2.26%	586.89
Ironman	116	166	8	4.82%	4.82%	4.82%	32.00	5.38%	933.73
Jump	122	122	17	13.93%	13.93%	13.93%	61.30	5.96%	819.67
Jumping	313	313	47	15.02%	15.02%	15.02%	32.48	51.32%	760.38
Liquor	1741	1741	1436	82.48%	82.48%	82.48%	66.01	9.51%	168.87
Matrix	100	100	35	35.00%	35.00%	35.00%	31.77	6.85%	600.00
MotorRolling	164	164	30	18.29%	18.29%	18.29%	31.83	6.31%	792.68
Panda	3000	2730	967	32.23%	35.42%	33.75%	32.73	78.22%	628.57
RedTeam	1918	1918	1918	100.00%	100.00%	100.00%	64.65	73.33%	0.00
Shaking	365	365	364	99.73%	99.73%	99.73%	32.20	73.05%	0.00
Singer2	366	366	366	100.00%	100.00%	100.00%	65.29	56.56%	0.00
Skating1	400	400	360	90.00%	90.00%	90.00%	64.76	30.51%	60.00
Skating2	473	473	332	70.19%	70.19%	70.19%	65.82	3.23%	190.27
Skiing	81	81	6	7.41%	7.41%	7.41%	31.21	21.77%	901.23
Soccer	392	392	90	22.96%	22.96%	22.96%	32.54	72.14%	706.63
Surfer	376	376	376	100.00%	100.00%	100.00%	32.62	71.95%	0.00
Sylvester	1345	1345	1345	100.00%	100.00%	100.00%	31.66	54.70%	0.00
Tiger2	365	365	364	99.73%	99.73%	99.73%	32.32	68.76%	0.00
Trellis	569	569	569	100.00%	100.00%	100.00%	32.25	42.66%	0.00
Walking	412	412	380	92.23%	92.23%	92.23%	66.68	26.75%	0.00
Walking2	500	500	211	42.20%	42.20%	42.20%	60.33	11.59%	502.00
Woman	597	597	102	17.09%	17.09%	17.09%	65.03	8.88%	643.22
Pedestrian2	338	266	35	10.36%	13.16%	11.59%	65.50	12.30%	864.66
Pedestrian3	184	156	52	28.26%	33.33%	30.59%	65.23	80.21%	608.97
Car	945	860	860	91.01%	100.00%	95.29%	64.59	0.01%	0.00
Motorcross	2665	1412	3	0.11%	0.21%	0.15%	32.02	30.69%	881.73
Volkswagen	3923	5141	2711	69.11%	52.73%	59.82%	23.27	5.20%	473.84
Carchase	2813	8660	351	12.48%	4.05%	6.12%	22.86	37.34%	959.35
Average	720.13	818.44	443.55	62.81%	62.72%	62.69%	46.95	37.34%	336.34

Table 5 Tracking results for TLD

Video	Response	ON	TP	*P*	*R*	*F*	FR	AOP	ANF
Basketball	225	725	20	8.89%	2.76%	4.21%	7.52	1.54%	965.52
Biker	65	142	65	100.00%	45.77%	62.80%	12.62	33.97%	542.25
Bird1	1	408	1	100.00%	0.25%	0.49%	15.88	0.24%	997.55
BlurBody	202	334	202	100.00%	60.48%	75.37%	12.26	41.75%	395.21
BlurCar2	435	585	435	100.00%	74.36%	85.29%	10.60	64.84%	256.41
BlurFace	493	493	493	100.00%	100.00%	100.00%	6.37	66.18%	0.00
BlurOwl	320	631	320	100.00%	50.71%	67.30%	14.45	40.90%	492.87
Bolt	5	350	5	100.00%	1.43%	2.82%	6.87	1.22%	985.71
Box	1033	1161	720	69.70%	62.02%	65.63%	6.06	35.99%	235.14
Car1	501	1020	314	62.67%	30.78%	41.29%	15.67	17.22%	645.10
Car4	309	659	232	75.08%	47.93%	47.93%	13.86	25.19%	531.11
CarDark	393	393	212	53.94%	53.94%	53.94%	15.29	22.33%	335.88
CarScale	170	252	158	92.94%	62.70%	74.88%	10.44	7.76%	353.17
ClifBar	201	472	201	100.00%	42.58%	59.73%	14.01	38.43%	574.15
Couple	90	140	74	82.22%	52.86%	64.35%	15.08	19.18%	457.14
Crowds	347	347	69	19.88%	19.88%	19.88%	8.37	7.76%	789.63
David	761	761	761	100.00%	100.00%	100.00%	11.65	69.61%	0.00
Deer	43	71	29	67.44%	40.85%	50.88%	12.02	28.51%	591.55
Diving	189	215	46	24.34%	21.40%	22.77%	7.58	10.69%	493.02
DragonBaby	35	113	30	85.71%	26.55%	40.54%	9.04	21.58%	734.51
Dudek	1145	1145	1028	89.78%	89.78%	89.78%	5.60	60.22%	0.00
Football	362	362	287	79.28%	79.28%	79.28%	6.80	50.09%	151.93
Freeman4	171	283	83	48.54%	29.33%	36.56%	15.10	11.56%	628.98
Girl	475	500	390	82.11%	78.00%	80.00%	10.88	49.55%	90.00
Human3	128	1698	16	12.50%	0.94%	1.75%	7.47	0.70%	987.04
Human4	667	667	376	56.37%	56.37%	56.37%	5.38	28.32%	401.80
Human6	556	792	461	82.91%	58.21%	68.40%	5.03	35.95%	392.68
Human9	146	305	49	33.56%	16.07%	21.73%	11.10	11.87%	780.33
Ironman	4	166	4	100.00%	2.41%	4.71%	9.06	2.15%	975.90
Jump	4	122	4	100.00%	3.28%	6.35%	15.68	9.60%	967.21
Jumping	247	313	247	100.00%	78.91%	88.21%	14.14	58.36%	210.86
Liquor	1089	1741	1007	92.47%	57.84%	71.17%	7.21	50.32%	407.24
Matrix	1	100	1	100.00%	1.00%	1.98%	6.03	0.42%	990.00
MotorRolling	26	164	26	100.00%	15.85%	27.37%	14.82	14.37%	841.46
Panda	1396	2730	1080	77.36%	39.56%	52.35%	18.25	21.58%	530.40
RedTeam	1536	1918	1332	86.72%	69.45%	77.13%	6.17	44.04%	227.84
Shaking	132	365	66	50.00%	18.08%	26.56%	13.19	12.15%	778.08
Singer2	11	366	10	90.91%	2.73%	5.31%	10.35	2.49%	972.68
Skating1	192	400	166	86.46%	41.50%	56.08%	10.12	18.89%	575.00
Skating2	343	473	9	2.62%	1.90%	2.21%	9.18	2.02%	904.86
Skiing	26	81	6	23.08%	7.41%	11.22%	11.33	6.49%	925.93
Soccer	27	392	27	100.00%	6.89%	12.89%	12.01	12.69%	931.12
Surfer	376	376	369	98.14%	98.14%	98.14%	11.74	57.01%	5.32
Sylvester	1345	1345	1213	90.19%	90.19%	90.19%	6.47	57.16%	44.61
Tiger2	170	365	132	77.65%	36.16%	49.35%	11.68	20.29%	569.86
Trellis	235	569	190	80.85%	33.39%	47.26%	15.61	28.47%	644.99
Walking	412	412	384	93.20%	93.20%	93.20%	5.54	58.25%	2.43
Walking2	497	500	471	94.77%	94.20%	94.48%	11.71	55.23%	22.00
Woman	392	597	231	58.93%	38.69%	46.71%	13.08	22.16%	584.59
Pedestrian2	270	266	254	94.07%	95.49%	94.78%	19.50	38.11%	293.23
Pedestrian3	148	156	147	99.32%	94.23%	94.23%	11.63	65.21%	0.00
Car	894	860	854	95.53%	99.30%	97.38%	11.23	77.94%	17.44
Motorcross	720	1412	696	96.67%	49.29%	65.29%	18.33	33.45%	542.49
Volkswagen	6187	5141	4847	78.34%	94.28%	85.58%	12.10	12.04%	592.49
Carchase	1120	8660	1119	99.91%	12.92%	22.88%	16.45	17.43%	854.62
Average	495.78	818.44	399.44	78.09%	46.94%	52.67%	11.19	37.58%	513.12

Table 6 An overall comparison of the proposed method, KCF, and TLD

Method	Response	ON	TP	*P*	*R*	*F*	FR	AOP	ANF
Proposed	557.31	818.44	483.31	84.63%	58.97%	63.48%	22.94	41.84%	388.90
KCF	720.13	818.44	443.55	62.81%	62.72%	62.69%	46.95	37.34%	336.34
TLD	495.78	818.44	399.44	78.09%	46.94%	52.67%	11.19	37.58%	513.12

since the proposed RPA can improve the tracking step and the utilization of the Kalman filter can improve the detection step, as described in Section 2, the learning step is affected positively by these improvements. Consequently, the proposed method shows better recall than the TLD method in most of the videos.

For the f-measure results, as defined in Eq. (8), f-measure is a measurement of precision and recall. Since the proposed method outperformed TLD and KCF by precision, the f-measure for the proposed method is 0.79% and 10.81% higher than that of KCF and TLD, respectively (Table 6 and Fig. 13).

Regarding AOP, as shown in Tables 3–6, the performance of the proposed method is 4.5% and 4.26% higher than that of KCF and TLD, respectively. Note that the average AOP was calculated using the method illustrated in Fig. 7 and all the response resulting bounding boxes of the tracking were used for calculation. Since the KCF method generated more false detections than the proposed method and TLD, AOP was decreased by this false detection. For ANF, the proposed method outperformed TLD by 124.21 frames (Table 6). As analyzed before, KCF achieved better recall than the proposed method due to its simple and effective framework. As a result, ANF of KCF is better than that of the proposed method by 52.57 frames (Table 6).

For FR, as shown in Table 6, since the KCF tracking framework is simpler than the frameworks for TLD and the proposed method, KCF obtained a faster speed than TLD and the proposed method. For the proposed method, due to the use of a Kalman filter, the detection step was greatly accelerated; as a result, the FR of the proposed method is 1.05 times higher than that of TLD.

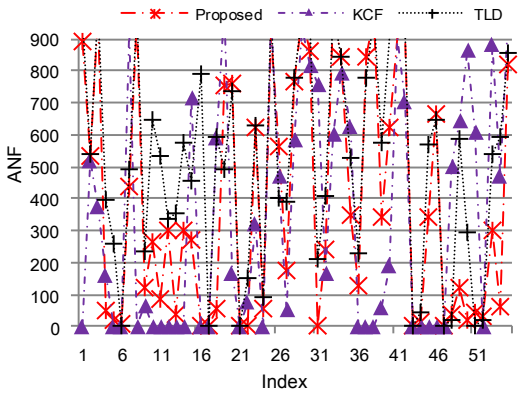

Fig. 13 ANF for TLD, KCF, and the proposed method

5 Conclusions

Long-term visual tracking is a challenging task in the field of computer vision. The difficulties in long-term tracking include illumination change, image noise, occlusions, and change in the object size or its appearance. The aim of this study is to develop a robust long-term object-tracking framework to overcome these challenges by improving the performance in TLD. To do this, a reliable point assignment algorithm based on the wavelet transform is proposed. By searching the local maximal wavelet coefficients, the reliable points located by the proposed method are rich with information and discriminant for object tracking. Therefore, the proposed method is robust against image illumination change, image noise, and object appearance deformation, leading to an improvement in the tracking step. To speed up the detection step and reduce the false detection results, the Kalman filter is applied to integrate with the TLD framework. RPA improves the tracking precision and the Kalman filter reduces the false detections. Consequently, the learning step is improved by the tracking and detection step, which feeds back to the detection step positively. As a result, compared with the TLD and KCF approaches, the new framework obtains better tracking precision, f-measure, and AOP. Since KCF tracks objects by training and detection, the KCF processing speed is much higher than those of the proposed method and TLD. However, the simple KCF framework makes it generate higher false detection than the other two methods do. Moreover,

the simple KCF framework makes it fail to track those targets that disappear from the video and reappear.

Acknowledgements

The authors thank Professor Sheng-ming JIANG for his good advice.

References

Bay, H., Ess, A., Tuytelaars, T., *et al*., 2008. Speeded-up robust features (SURF). *Comput. Vis. Image Understand.*, **110**(3):346-359. http://dx.doi.org/10.1016/j.cviu.2007.09.014

Brox, T., Bruhn, A., Papenberg, N., *et al*., 2004. High accuracy optical flow estimation based on a theory for warping. European Conf. on Computer Vision, p.25-36. http://dx.doi.org/10.1007/978-3-540-24673-2_3

Cheng, C.W., Ou, W.L., Fan, C.P., 2016. Fast ellipse fitting based pupil tracking design for human-computer interaction applications. IEEE Int. Conf. on Consumer Electronics, p.445-446. http://dx.doi.org/10.1109/ICCE.2016.7430685

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.886-893. http://dx.doi.org/10.1109/CVPR.2005.177

Elhamod, M., Levine, M.D., 2013. Automated real-time detection of potentially suspicious behavior in public transport areas. *IEEE Trans. Intell. Transp. Syst.*, **14**(2): 688-699. http://dx.doi.org/10.1109/TITS.2012.2228640

Elmenreich, W., Koplin, M.A., 2011. Time-triggered object tracking subsystem for advanced driver assistance systems. *Elektrotechn. Inform.*, **128**(6):203-208. http://dx.doi.org/10.1007/s00502-011-0004-x

Gonzalez, R.C., Woods, R.E., 2002. Digital Image Processing (2nd Ed.). Prentice Hall, Inc., New Jersey.

Harris, C., Stephens, M., 1988. A combined corner and edge detector. Proc. Alvey Vision Conf., p.147-151. http://dx.doi.org/10.5244/C.2.23

Henriques, J.F., Caseiro, R., Martins, P., *et al*., 2015. High-speed tracking with kernelized correlation filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, **37**(3):583-596. http://dx.doi.org/10.1109/TPAMI.2014.2345390

Jeong, J.M., Yoon, T.S., Park, J.B., 2014. Kalman filter based multiple objects detection-tracking algorithm robust to occlusion. Proc. SICE Annual Conf., p.941-946. http://dx.doi.org/10.1109/SICE.2014.6935235

Jia, C.X., Wang, Z.L., Wu, X., *et al*., 2015. A tracking-learning-detection (TLD) method with local binary pattern improved. IEEE Int. Conf. on Robotics and Biomimetics, p.1625-1630. http://dx.doi.org/10.1109/ROBIO.2015.7419004

Jung, Y., Yoon, Y., 2015. Behavior tracking model in dynamic situation using the risk ratio EM. Int. Conf. on Information Networking, p.444-448. http://dx.doi.org/10.1109/ICOIN.2015.7057942

Kalal, Z., Mikolajczyk, K., Matas, J., 2010a. Forward-backward error: automatic detection of tracking failures. 20th Int. Conf. on Pattern Recognition, p.23-26. http://dx.doi.org/10.1109/ICPR.2010.675

Kalal, Z., Matas, J., Mikolajczyk, K., 2010b. P-N learning: bootstrapping binary classifiers by structural constraints. IEEE Conf. on Computer Vision and Pattern Recognition, 49-56. http://dx.doi.org/10.1109/CVPR.2010.5540231

Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, **34**(7):1409-1422. http://dx.doi.org/10.1109/TPAMI.2011.239

Kalman, R.E., 1960. A new approach to linear filtering and prediction problems. *J. Basic Eng.*, **82**(1):35-45. http://dx.doi.org/10.1115/1.3662552

Kaur, H., Sahambi, J.S., 2015. Vehicle tracking using fractional order Kalman filter for non-linear system. Int. Conf. on Computing, Communication and Automation, p.474-479. http://dx.doi.org/10.1109/CCAA.2015.7148423

Kong, H., Akakin, H.C., Sarma, S.E., 2013. A generalized Laplacian of Gaussian filter for blob detection and its applications. *IEEE Trans. Cybern.*, **43**(6):1719-1733. http://dx.doi.org/10.1109/TSMCB.2012.2228639

Li, Y., Zhu, J.K., Hoi, S.C.H., 2015. Reliable patch trackers: robust visual tracking by exploiting reliable patches. IEEE Conf. on Computer Vision and Pattern Recognition, p.353-361. http://dx.doi.org/10.1109/CVPR.2015.7298632

Liu, S., Zhang, T.Z., Cao, X.C., *et al*., 2016. Structural correlation filter for robust visual tracking. IEEE Conf. on Computer Vision and Pattern Recognition, p.4312-4320. http://dx.doi.org/10.1109/CVPR.2016.467

Liu, T., Wang, G., Yang, Q.X., 2015. Real-time part-based visual tracking via adaptive correlation filters. IEEE Conf. on Computer Vision and Pattern Recognition, p.4902-4912. http://dx.doi.org/10.1109/CVPR.2015.7299124

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, **60**(2):91-110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

Ning, G.H., Zhang, Z., Huang, C., *et al*., 2016. Spatially supervised recurrent convolutional neural networks for visual object tracking. arXiv:1607.05781v1.

Prakash, U.M., Thamaraiselvi, V.G., 2014. Detecting and tracking of multiple moving objects for intelligent video surveillance systems. 2nd Int. Conf. on Current Trends in Engineering and Technology, p.253-257. http://dx.doi.org/10.1109/ICCTET.2014.6966297

Redmon, J., Divvala, S., Girshick, R., *et al.*, 2016. You only look once: unified, real-time object detection. IEEE Conf. on Computer Vision and Pattern Recognition, p.779-788. http://dx.doi.org/10.1109/CVPR.2016.91

Sun, X., Yao, H.X., Zhang, S.P., 2010. A refined particle filter method for contour tracking. *SPIE*, **7744**:77441M. http://dx.doi.org/10.1117/12.863450

Tarkov, M.S., Dubynin, S.V., 2013. Real-time object tracking by CUDA-accelerated neural network. *J. Comput. Sci. Appl.*, **1**(1):1-4. http://dx.doi.org/10.12691/jcsa-1-1-1

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.511-518. http://dx.doi.org/10.1109/CVPR.2001.990517

Xu, F., Gao, M., 2010. Human detection and tracking based on HOG and particle filter. 3rd Int. Congress on Image and Signal Processing, p.1503-1507. http://dx.doi.org/10.1109/CISP.2010.5646273

Yu, H.M., Zeng, X., 2015. Visual tracking combined with ranking vector SVM. *J. Zhejiang Univ.* (*Eng. Sci.*), **49**(6): 1015-1021 (in Chinese). http://dx.doi.org/10.3785/j.issn.1008-973X.2015.06.003

Yu, W.S., Tian, X.H., Hou, Z.Q., *et al.*, 2015. Multi-scale mean shift tracking. *IET Comput. Vis.*, **9**(1):110-123. http://dx.doi.org/10.1049/iet-cvi.2014.0077

Zhang, R.F., Xiao, H.H., Deng, T., *et al*., 2016. A robust point detection algorithm based on wavelet transform for visual tracking. Int. Congress on Image and Signal Processing, Biomedical Engineering and Informatics, p.1-5. http://dx.doi.org/10.1109/CISP-BMEI.2016.7852672