Adaptive Learning Compressive Tracking Based on Kalman Filter

Xingyu Zhou, Dongmei Fu(✉), Yanan Shi, and Chunhong Wu

School of Automation and Electrical Engineering,
University of Science and Technology Beijing, Beijing, China
fdm_ustb@ustb.edu.cn

Abstract. Object tracking has theoretical and practical application value in video surveillance, virtual reality and automatic navigation. Compressive tracking(CT) is widely used because of its advantages in accuracy and efficiency. However, the compressive tracking has the problem of tracking drift when there are object occlusion, abrupt motion and blur, similar objects. In this paper, we propose adaptive learning compressive tracking based on Kalman filter (ALCT-KF). The CT is used to locate the object and the classifier parameter can be updated adaptively by the confidence map. When the heavy occlusion occurs, Kalman filter is used to predict the location of object. Experimental results show that ALCT-KF has better tracking accuracy and robustness than current advanced algorithms and the average tracking speed of the algorithm is 39 frames/s, which can meet the requirements of real-time.

Keywords: Kalman filter · Compressive tracking · Confidence map
Adaptive learning

1 Introduction

Object tracking has been widely used in video surveillance and robotics which is a very popular topic in computer vision [1]. In recent years, although many tracking methods have been proposed and much success has been demonstrated, robust tracking is still a challenging task due to factors such as occlusion, fast moving, motion blur and pose change [2].

In order to deal with these factors, how to build an effective adaptive appearance model is particularly important. In general, tracking algorithms can be categorized into two classes: generative and discriminative algorithms [3]. The generative tracking algorithms aim at modeling the target and finding the location of the target by searching the image blocks which are the most similar to the target model. Kumar et al. [4] combine Kalman filter with geometric shape template matching method and can solve multi-target segmentation and merging. Zhan et al. [5] propose a combination of mean shift algorithm and Kalman filter tracking algorithm which can avoid the model update error. Wang et al. [6] use partial least squares (PLS) to study low dimensional distinguishable subspaces, and the tracking drift problem is alleviated by online update of the apparent model. Hu et al. [7] introduce the sparse weight constraint to dynamically select the relevant template in the global template set and use the multi-feature joint

Y. Zhao et al. (Eds.): ICIG 2017, Part III, LNCS 10668, pp. 243–253, 2017.
https://doi.org/10.1007/978-3-319-71598-8_22

sparse representation for multi-target tracking under occlusion. However, the above generative algorithms ignore the background information. When the other object with similar texture to the target or the object is occluded, the tracking algorithm is easily interrupted or tracking failure.

Discriminative tracking algorithms consider that target tracking as a two element classification problem, the purpose is to find a boundary that divide the target from the background [8]. Babenko et al. [9] propose a multiple instance learning (MIL) approach, because of its feature selection calculation complexity, it leads to poor real-time performance. Kaur and Sahambi et al. [10] propose an improved steady-state gain Kalman filter. By introducing a fractional feedback loop in the Kalman filter, the proposed algorithm solves the problem of abrupt motion. Zhang et al. [11] make full use of hybrid SVMs for appearance models to solve the blurring problem of the former background boundary and avoid the drift problem effectively. But these discriminative methods involve high computational cost, which hinders their real-time applications.

In order to take advantage of the above two kinds of methods, this paper proposes an adaptive learning compressive tracking algorithm [12, 13] based on Kalman filter (ALCT-KF), which is used to solve the problems of severe occlusion, fast moving, similar object and illumination change. The adaptive learning compressive tracking algorithm uses CT algorithm to track the target, and calculate the value of the Peak-to-Sidelobe (PSR) by confidence map to update Bayesian classifier adaptively. When the *PSR* is less than a certain threshold, the object is considered to be heavy occlusion, then uses the Kalman filter to predict the location of object.

The rest of this paper is organized as follows. Section 2 gives a brief review of original CT. The proposed algorithm is detailed in Sect. 3. Section 4 shows the experimental results of proposed and we conclude in Sect. 5.

2 Compressive Tracking

As shown in [12, 13] are based on compressive sensing theory, a very sparse random matrix is adopted that satisfies the restricted isometry property (RIP), facilitating projection from the high-dimensional Haar-like feature vector to a low-dimensional measurement vector

$$V = Rx, \tag{1}$$

where $R \in R^{n\times m}$ $(n \ll m)$ is sparse random matrix, feature vector $x \in R^{m\times 1}$, compressive feature vector $V \in R^{n\times 1}$.

$$R(i,j) = r_{i,j} = \sqrt{s} \times \begin{cases} 1 & \text{with probability } \dfrac{1}{2s} \\ 0 & \text{with probability } 1 - \dfrac{1}{s} \\ -1 & \text{with probability } \dfrac{1}{2s} \end{cases} \tag{2}$$

where $s = m/(a\log_{10}(m))$, $m = 10^6 \sim 10^{10}$, $a = 0.4$. R becomes very sparse, and the number of non-zero elements for each row is only 4 at most, further reducing the computational complexity.

The compressed features v are obtained by (1) and (2) which inputs to the naive Bayesian classifier and the position of the target is determined by the response value. Assuming that the elements in v are independently distributed, the naive Bayesian classifier is constructed:

$$\begin{aligned} H(v) &= \log\left(\frac{\prod_{i=1}^{n} p(v_i \mid y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i \mid y=0)p(y=0)}\right) \\ &= \sum_{i=1}^{n} \log\left(\frac{p(v_i \mid y=1)}{p(v_i \mid y=0)}\right) \end{aligned} \tag{3}$$

where $p(y=1) = p(y=0) = 0.5$, and $y \in \{0, 1\}$ is binary variable which represents the sample label. The conditional distributions $p(v_i \mid y=1)$ and $p(v_i \mid y=0)$ in $H(v)$ are assumed to be Gaussian distributed with four parameters $(\mu_i^1, \delta_i^1, \mu_i^0, \delta_i^0)$

$$p(v_i \mid y=1) \sim N(\mu_i^1, \sigma_i^1),\ p(v_i \mid y=0) \sim N(\mu_i^0, \sigma_i^0) \tag{4}$$

where $\mu_i^1(\mu_i^0)$ and $\mu_i^0(\delta_i^0)$ are mean and standard deviation of the positive (negative) class. These parameters can be updated by

$$\begin{aligned} \mu_i^1 &\leftarrow \lambda\mu^1 + (1-\lambda)\mu_i^1 \\ \sigma_i^1 &\leftarrow \sqrt{\lambda(\sigma^1)^2 + (1-\lambda)(\sigma_i^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \end{aligned} \tag{5}$$

where λ is the learning parameter, and

$$\begin{aligned} \sigma^1 &= \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1} (v_i(k) - \mu^1)^2} \\ \mu^1 &= \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k) \end{aligned} \tag{6}$$

Negative sample parameters μ_i^0 and σ_i^0 are updated with the similar rules.

Compressive tracking is simple and efficient, but the problem still exits: the classifier is updated by Eq. (5) which uses a fixed learning rate λ. When the occlusion and other conditions occur, it may cause the classifier update error.

3 Proposed Algorithm

3.1 Adaptive Learning Compressive Tracking (ALCT)

Compressive tracking algorithm is difficult to re-find the right object when it tracks drift or failure. One of the main reasons is that $p(v_i \mid y=0)$ and $p(v_i \mid y=1)$ are determined

by the four parameters μ_i^0, μ_i^1, σ_i^0, σ_i^1 while a fixed learning parameter λ is used in Eq. (5). When the occlusion or other conditions occur, λ may cause the classifier to update incorrectly.

According to Eq. (3), we define the non-linear function for the naive Bayes classifier $H(v)$ as objective confidence

$$c(x) = p(y = 1 \mid x) = \sigma(H(v)) \tag{7}$$

where $\sigma(\cdot)$ is a sigmoid function, $\sigma(x) = (1/1 + e^{-x})$.

The Peak-to-Sidelobe(PSR) [14], which measures the strength of a correlation peak, can be used to detect occlusions or tracking failure.

$$PSR(t) = \frac{\max(c_t(x)) - \mu_t}{\sigma_t} \tag{8}$$

where $c_t(x)$ denotes the classifier response value for all the candidate positions at the t-th frame, split into the peak which is the maximum value $\max(c_t(x))$ and the sidelobe which is the rest of the search position excluding an 11×11 window around the peak. μ_t and σ_t are the mean and standard deviation of the sidelobe. Taking the Cliff bar sequence as an example, the *PSR* distribution is shown in Fig. 1.

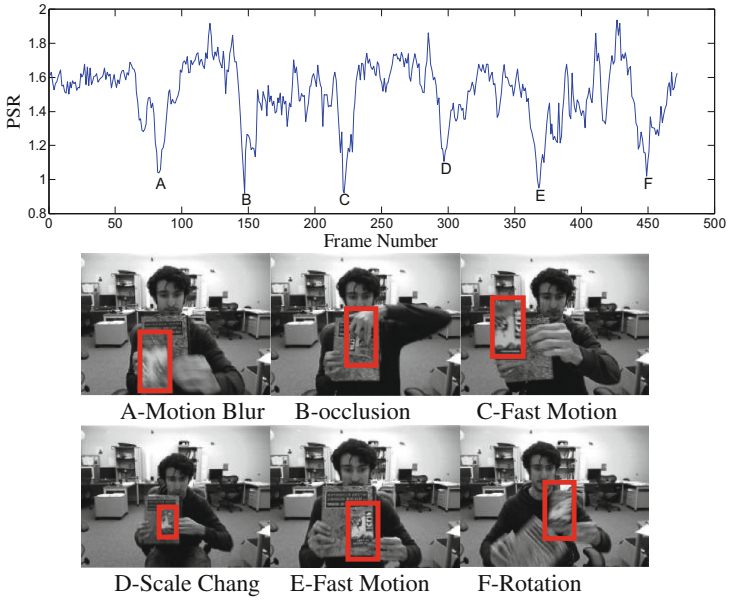


Fig. 1. Analysis of *PSR* in sequence of Cliff bar

Figure 1 shows the *PSR* can locate the most challenging factors of that video. In the first 75 frames, object has few interference factors and *PSR* stabilized at about 1.6. The object moves fast and causes the target area to be blurred, PSR is down to point A in

the 75–90 frames. When there is no longer moving blur, *PSR* gradually returns to the normal level. In the same way, when the object undergoes occlusion, fast motion, scale change, rotation, which cause the values of PSR down to the valley point, corresponding to B, C, D, E, F respectively in Fig. 1. The value of *PSR* can reflect the influence of factors. The higher *PSR* is, the higher confidence of target location. Therefore, when the PSR is less than a certain threshold, the classifier should be updated with a smaller learning rate, which can improve the anti-interference ability of the model.

Experiments (see Fig. 1) show that when the value of *PSR* is higher than 1.6, the tracking results are completely credible. If *PSR* is less than 1.6, the object may be occlusion, pose and illumination change. So we can determine the update weight of the classifier according to the *PSR* of each frame. The new update formula is shown in Eq. (9):

$$\begin{cases} w_t = \begin{cases} 0 & PSR_t < PSR_0 \\ \exp[-(PSR_t - PSR_1)^2] & PSR_0 < PSR_t < PSR_1 \\ 1 & \text{other} \end{cases} \\ \mu_i^1 \leftarrow (1-\lambda w_t)\mu_i^1 + \lambda w_t \mu^1 \\ \sigma_i^1 \leftarrow \sqrt{(1-\lambda w_t)(\sigma_i^1)^2 + \lambda w_t(\sigma^1)^2 + \lambda w_t(1-\lambda w_t)(\mu_i^1 - \mu^1)^2} \end{cases} \tag{9}$$

where PSR_t represents the *PSR* at the t-th frame, PSR_0 and PSR_1 are two thresholds. When $PSR_0 < PSR_t < PSR_1$, it is considered that the object may undergo partial occlusion, fast motion, pose change. When $PSR_t < PSR_0$, it is considered that the object is completely occluded, and classifier is not updated at this time. At this time, Kalman filter is used to predict the position of object.

3.2 Heavy Conclusion

In the process of object tracking, occlusion, illumination change, fast moving and similar target can not be avoided. If the above factors occur, the accuracy of many algorithms are obviously decreased. The adaptive learning compressive tracking algorithm proposed in this paper can meet the factors of partial occlusion and slow illumination change, but it needs to improve the algorithm for heavy occlusion.

Kalman filter algorithm [15] is mainly used to estimate the target location and state. The algorithm uses the position and velocity of the object as the state vector to describe the change of the object state. Kalman filter algorithm can also effectively reduce the influence of noise in the object tracking process. The state equations and the observation equation of the Kalman filter are as follows:

$$x_{t+1} = \phi x_t + w_t \tag{10}$$

$$Z_t = Hx_t + V_t \tag{11}$$

where $x_t(x_{t+1})$ is the state vector of the $t(t+1)$ moment, Z_t is the observation vector of the t moment, ϕ is state transition matrix, H is observation matrix, w_t is state noise vector of system disturbance, v_t is observed noise vector.

In Sect. 3.1, it is proved when $PSR_t < PSR_0$ considers the target to be heavy occlusion. Because the Kalman filter can predict the position of the target in the next frame and effectively reduce the influence of noise, we use it to solve the above problems.

Kalman filter can be divided into two phases: prediction and updating. The prediction phase is mainly based on the state of the current frame target to estimate the state of the next frame. In the update phase, the estimate of the prediction phase is optimized by using the observations of the next frame to obtain more accurate new predictions. Assuming that the target has a serious occlusion at $(t + 1)$-th frame, the Kalman filter is used to re-estimate the object position.

(1) prediction phase

State prediction equation:

$$x_{t+1}^{-} = \phi x_t^{+} \tag{12}$$

where x_t^{+} is the tracking result of the ALCT algorithm at the t-th frame.

Error covariance prediction equation:

$$P_{t+1}^{-} = \phi P_t^{+} \phi^T + Q \tag{13}$$

where P_t^{+} is the covariance matrix at t frame, Q is the state noise covariance matrix and the value is constant.

(1) updating phase

Gain equation:

$$K_{t+1} = P_{t+1}^{-} H^T (H P_t^{-} H^T + R)^{-1} \tag{14}$$

where K_{t+1} is the Kalman gain matrix, R is the measurement noise covariance matrix and the value is constant.

Error covariance modification equation:

$$P_{t+1}^{+} = (1 - K_{t+1} H) P_{t+1}^{-} \tag{15}$$

State modification equation:

$$x_{t+1}^{+} = x_{t+1}^{-} + K_{t+1}(Z_{t+1} - H x_{t+1}^{-}) \tag{16}$$

where Z_{t+1} is the object position that the ALCT algorithm tracks when it is at $(t + 1)$-th frame, x_{t+1}^{+} is the position of the estimated object at the $(t + 1)$-th frame.

Then, using the ALCT algorithm to track the position of object at the t-th frame. If $PSR_{t+2} < PSR_0$ then re-estimate the target in the frame position by Eqs. (12)–(16), otherwise, ALCT is used to track the object at next frame. The flow chart of adaptive learning compressive tracking algorithm based on Kalman filter (ALCT-KF) is shown in Fig. 2.

Firstly, the position of the target in the first frame is manually calibrated and the object is tracked by the ALCT algorithm. Then, the PSR is calculated by the target confidence map and the Bayesian classifier is updated by the PSR. If the PSR is less

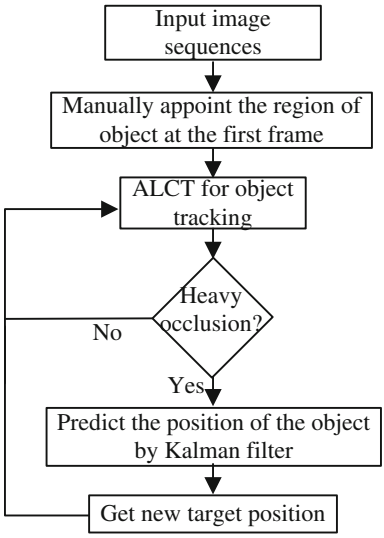


Fig. 2. Flow of ALCT-KF algorithm

than a certain threshold, it is considered that the target is serious occlusion. At this time, the Kalman filter is used to predict the position of the target in the current frame and the predicted position is assigned to ALCT algorithm for object tracking in next frame.

4 Experiment

In order to validate the proposed algorithm, 6 different challenging video sequences are adopted, including occlusion, illumination, pose change, fast motion, and similar object. We compare the proposed ALCT-KF algorithm with the state-of-art methods, Compressive tracking(CT) [13], Online discriminative feature selection(ODFS) [16], Spatio-temporal context learning(STC) [17], Tracking-Learning-Detection(TLD) [18]. Implemented in MATLAB2013a, Core(TM)i5-4570CPU and 4 GB RAM. As is shown in Fig. 1, the thresholds of Eq. (9) are set to $PSR_0 \in [1.2,\ 1.4]$ and $PSR_1 \in [1.6 - 1.8]$. The Kalman filter parameter is set to:

$$\varphi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 0.2845 & 0.0045 \\ 0.0045 & 0.0045 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 400 & 0 & 0 & 0 \\ 0 & 400 & 0 & 0 \\ 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 400 \end{bmatrix}$$

Table 1. Success rate (SR) (%) and average frames per second (FPS). (Top two result are shown in Bold and italic).

Sequences	ALCT-KF	CT	ODFS	STC	TLD
Dudek	**97.2**	*73.1*	71.2	61.6	72.6
FaceOcc2	*96.3*	86.2	**98.5**	93.0	70.2
Motocross	**92.4**	70.7	60.6	*77.8*	69.2
Cliff bar	**91.1**	82.0	*89.2*	86.4	63.1
David	*96.1*	89.8	93.1	**97.6**	92.9
Pedestrian	**83.1**	*63.0*	8.5	*2.5*	32.9
Average SR	**93.2**	*77.5*	70.2	69.8	66.8
Average FPS	39	*42*	38	**53**	16

Two metrics are used to evaluate the experimental results. (1) The first metric is the success rate which is defined as,

$$score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)} \tag{17}$$

where ROI_G is the ground truth bounding box and ROI_T is the tracking bounding box. If score is larger than 0.5 in one frame, the tracking result is considered as a success.

Table 1 shows the comparison of success rate in the test video. The proposed algorithm achieves the best or second best performance. Compared with the CT algorithm, the average tracking success rate of M-ALCT is improved by 15.7%. The last row of Table 1 gives the average frames per second. ALCT-KF performs well in speed (only slightly slower than CT method) which is faster than ODFS, TLD methods.

The second metric is the center location error which is defined as the euclidean distance between the central locations of the tracked object and the manually labeled ground truth.

$$CLE = \sqrt{(x_T - x_G)^2 + (y_T - y_G)^2} \tag{18}$$

The tracking error of the proposed algorithm is smaller than other algorithms, which can be maintained within 15 pixels (see Fig. 3).

The object in the Dudek and FaceOcc2 sequences (see Fig. 4) is subject to partial occlusion and heavy occlusion. In the Cliff bar and Motocross sequences (see Fig. 5), the object is abrupt motion and rotation which lead to the appearances of objects change significant and motion blur. The David and Pedestrian sequences (see Fig. 6) are challenging due to illumination variation and similar object. Through the above experiments the proposed algorithm effectively avoids the tracking failure when occlusion, abrupt motion, motion blur, similar target and other situations occur.

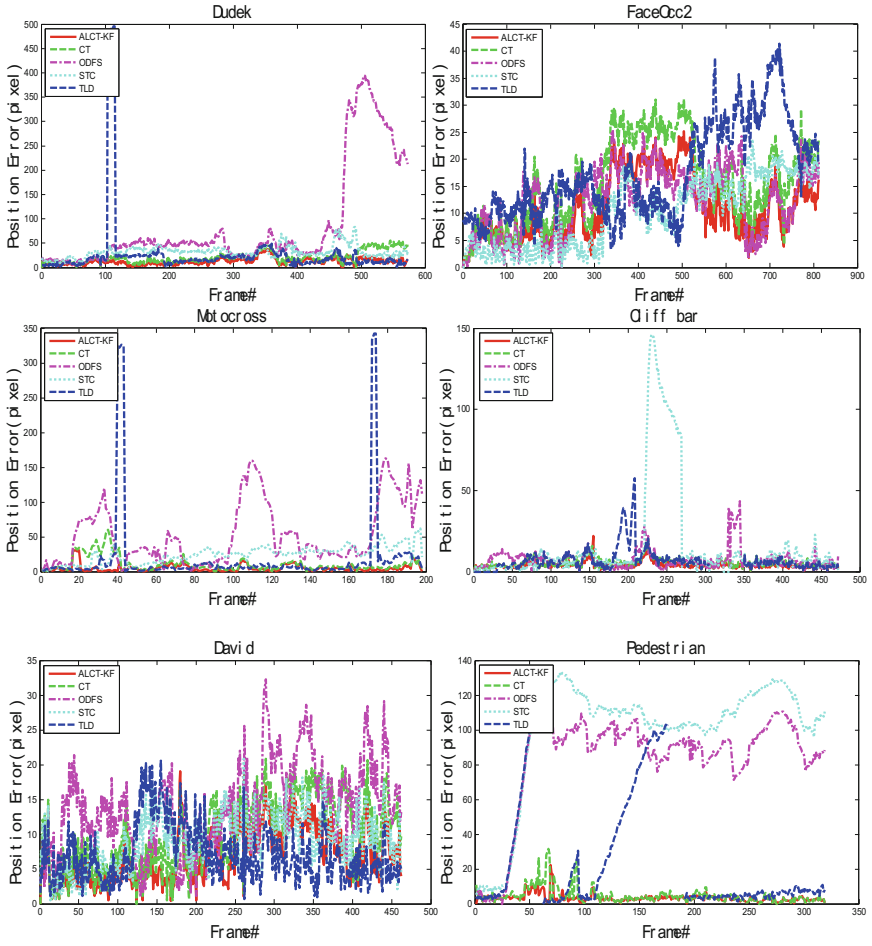


Fig. 3. Error plots in terms of center location error for 6 test sequences. (Color figure online)

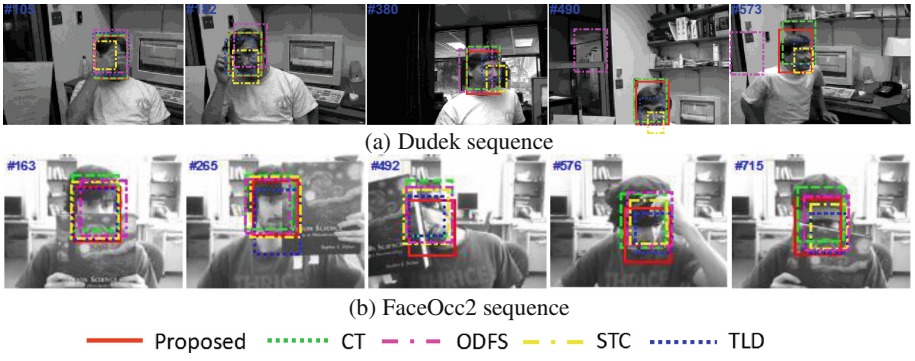
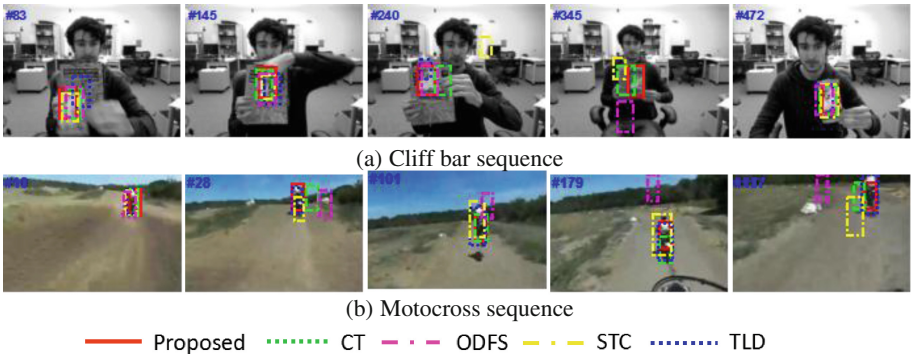


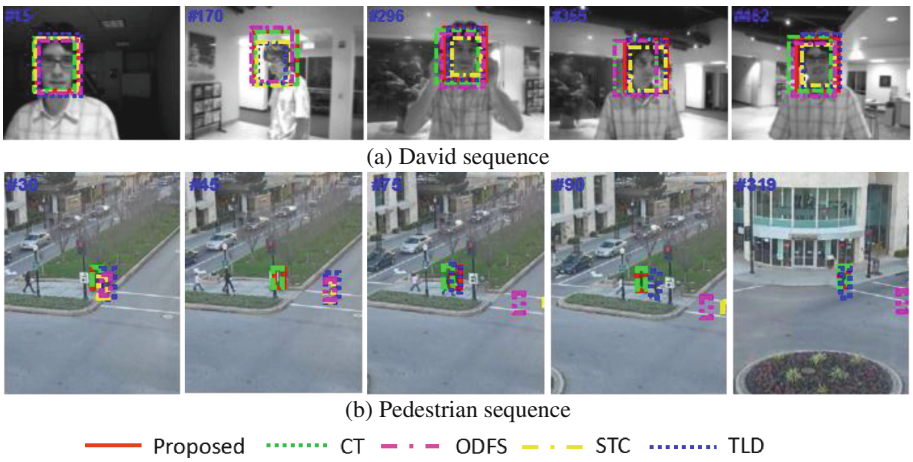
Fig. 4. Tracking results of the occlusion sequences. (Color figure online)



(a) Cliff bar sequence

(b) Motocross sequence

Fig. 5. Some sample tracking results of abrupt motion and rotation sequences. (Color figure online)



(a) David sequence

(b) Pedestrian sequence

Fig. 6. Tracking results of illumination variation and similar object. (Color figure online)

5 Conclusion

In this paper, an adaptive learning compressive tracking algorithm based on Kalman filter is proposed to deal with the problem of occlusion, fast moving, rotation and similar object. Tracking drift problem is alleviated by using tracking confidence map to adaptively update the classifier model and Kalman filter algorithm is used to predict the object location and reduce the impact of noise. Experiments show that the proposed algorithm has better tracking accuracy and robustness and it is easy to implement and achieves real-time performance.

References

1. Li, X., Hu, W., Shen, C., et al.: A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. **4**(4), 48–58 (2013)
2. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., et al.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2014)
3. Liu, W., Li, J., Chen, X., et al.: Scale-adaptive compressive tracking with feature integration. J. Electron. Imaging **25**(3), 3301801–3301810 (2016)
4. Kumar, P., Ranganath, S., Sengupta, K., et al.: Cooperative multitarget tracking with efficient split and merge handling. IEEE Trans. Circ. Syst. Video Technol. **16**(12), 1477–1490 (2006)
5. Zhan, J.P., Huang, X.Y., Shen, Z.X., et al.: Object tracking based on mean shift and kalman filter. J. Chongqing Inst. Technol. **3**, 76–80 (2010)
6. Wang, Q., Chen, F., Xu, W., et al.: Object tracking via partial least squares analysis. IEEE Trans. Image Process. **21**(10), 4454–4465 (2012)
7. Hu, W., Li, W., Zhang, X., et al.: Single and multiple object tracking using a multi-feature joint sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **37**(4), 816–833 (2015)
8. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, pp. 2411–2418 (2013)
9. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1619–1632 (2011)
10. Kaur, H., Sahambi, J.S.: Vehicle tracking in video using fractional feedback kalman filter. IEEE Trans. Comput. Imaging **2**(4), 550–561 (2016)
11. Zhang, S., Sui, Y., Yu, X., et al.: Hybrid support vector machines for robust object tracking. Pattern Recognit. **48**(8), 2474–2488 (2015)
12. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_62
13. Zhang, K., Zhang, L., Yang, M.H.: Fast compressive tracking. IEEE Trans. Pattern Anal. Mach. Intell. **36**(10), 2002–2015 (2014)
14. Bolme, D.S., Beveridge, J.R., Draper, B.A., et al.: Visual object tracking using adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, pp. 2544–2550 (2010)
15. Chen, S.Y.: Kalman filter for robot vision: a survey. IEEE Trans. Ind. Electron. **59**(11), 4409–4420 (2012)
16. Zhang, K., Zhang, L., Yang, M.H.: Real-time object tracking via online discriminative feature selection. IEEE Trans. Image Process. **22**(12), 4664–4677 (2013)
17. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H.: Fast visual tracking via dense spatio-temporal context learning. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 127–141. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_9
18. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)