Tracking an Object over 200 FPS with the Fusion of Prior Probability and Kalman Filter

Jun Chen
Cyber-Systems and Control,
Zhejiang University,
Hangzhou, Zhejiang, 310027, China
chenjun931206@163.com

Jin-Hui Zhao
Zhejiang University of Water Resources and Electric Power,
Hangzhou, Zhejiang, 310018, China
jhzhao2009@zju.edu.cn

Wei Zhang
Fair Friend Institute of ElectroMechanics,
Hangzhou Vocational & Technical College,
Hangzhou, Zhejiang, 310018,China
zhw618@zju.edu.cn

Yong Liu
Cyber-Systems and Control,
Zhejiang University,
Hangzhou, Zhejiang, 310027, China
yongliu@iipc.zju.edu.cn

ABSTRACT

Efficient object tracking is a challenge problem as it needs to distinguish the object by learned appearance model as quickly as possible. In this paper, a novel robust approach fusing the prediction information of Kalman filter and prior probability is proposed for tracking arbitrary objects. Firstly, we obtain an image patch based on predicted information by fusing the prior probability and Kalman filter. Secondly, the samples derived from the obtained image patch for our tracker are entered into support vector machine (SVM) to classify the object, where these samples need to be extracted features by Histogram of Oriented Gradients (HOG). Our approach has two advantages: efficient computation, and certain anti-interference ability. The samples obtained from image patch is less than that obtained from image, which makes SVM model more efficient in classification and reduces interference outside the image patch. Experimentally, we evaluate our approach on a standard tracking benchmark that includes 50 video sequences to demonstrate our tracker's nearly state-of-the-art performance compared with 5 trackers. Furthermore, because extracting samples and classifying HOG features is computationally very cheap, our tracker is much faster than these mentioned trackers. It achieves over 200 fps on the Intel i3 CPU for tracking an arbitrary object on benchmark.

CCS Concepts

• Computing methodologies→Tracking

Keywords

Object tracking; Support vector machines; Kalman filter; information fusion.

ICMLC 2020, February 15–17, 2020, Shenzhen, China

ACM ISBN 978-1-4503-7642-6/20/02…$15.00
DOI: https://doi.org/10.1145/3383972.3384011

1. INTRODUCTION

Object tracking is an important area of research in computer vision and has a wide range of applications such as automated video surveillance, automated vehicle navigation and autonomous driving, and person-following applications. Given some object of interest marked among consecutive frames of a video sequence, the task of "single-object tracking" is to locate this object in the corresponding video frames, despite dynamic background, diversity of appearance, morphological changes or other variations [1,2].

The trackers mentioned in this paper refer to the generic object trackers, which is the trackers that are not specialized for specific classes objects. The traditional trackers (trackers that do not use convolution neural networks (CNNs) to extract features) always use SVM, Random Forest classifiers, or correlation filters to learn a model of the object appearance [3-6]. The relatively good one is the Kernelized Correlation Filter (KCF) proposed by Joao F. Henriques et al. [7], which avoids matrix inversion by constructing a circulant matrix in the linear regression of the kernel space, thus enabling fast detection. Although the tracking speed of KCF can exceed 100 fps, the constructed circulant matrix is still too large which makes the computation less efficient. Zhang et al. [8] propose a real-time tracker by compressing low-dimensional subspace to retrain the information of high-dimensional image feature space. Babenko et al. [9] propose to speed up tracking through sample packet label posterior probability while ensuring tracking accuracy. Hare et al. [10] turn the tracking problem into a classification problem for the first time that is also the precursor to the work of KCF. The tracking speed of the two trackers above can reach over 10 fps. CNNs have been well applied in object tracking, because CNNs are naturally effective for extracting the features of objects. The biggest problem with the CNN-based trackers is that the tracking speed is very slow, usually only a few fps or a dozen fps on the GPU, because the process of training CNNs is very complicated.

In this work, we propose a more efficient tracking algorithm based the fusion of prior probability and Kalman filter that focus on improving the fps of object tracking. To achieve this goal, the first and foremost thing is to obtain an image patch of high confidence containing the object as the test sample set for the SVM classifier. Then, the SVM model is trained according to the marked object in the first frame and will not be trained in the subsequence frames. The smaller image patch of high confidence means that our test sample set is smaller, so the computational efficiency of SVM model is higher. In order to obtain such image patch, we use Kalman filter to estimate the motion state of object in the next frame based on the motion state of current frame, and the estimated object location is taken as the center of the image patch. Then, the high confidence interval obtained by prior probability is taken as the pixel length of the image patch. The SVM classifier can use this image patch as the test set to get a sample that is most similar to the object, and this sample is regarded as the object. At this point, the entire process with loop iteration completes the task of object tracking. Our tracker runs at 200+ fps on Intel i3 CPU that significantly outperforms state-of-the-art equivalents in terms of tracking speed on benchmark. The high speed of our tracker is critical for object tracking on computers or mobile devices with limited computational power.

2. PROPOSED TRACKER

How the proceed of our proposed approach is summarized in this section, which includes an organized review of all calculation steps and a performance test on Kalman filter.

2.1 Overview of the Proposed Tracking Algorithm

The program flow chat is shown in Fig.1, which makes out one cycle of the algorithm recursion and two main steps. After we initialize the parameters of algorithm and choose an arbitrary object on a video, the matrices and coordinate information of the picked-up object image will be calculated straight. The SVM model is trained by the HOG eigenvectors of positive and negative samples in the first frame, which is no longer trained in subsequent frames. And then, the Kalman filter and prior probability are used to estimate the motion state of the object in the next frame based on the motion state of the current frame, where the motion state includes displacement and velocity. Before obtaining the location of object in subsequent frames, SVM model classifies the candidate samples in the image patch, which sample features with highest probability of similarity to the object features in the candidate samples are used as the object in next frame. At last, we update the location of the object, and the entire process is repeated to achieve the effect of tracking.

2.2 Information Fusion

In this part, we choose the absolute value of the difference between ground truths and estimated values by kalman filter as the forecast error to predict the motion state of the object.

2.2.1 *Kalman Filter (KF)*

Kalman filter is called the optimal linear filter and has some distinctive advantages, such as simple implementation and time-domain calculation. Up to now, it has been applied to high-tech domain extensively, such as military, national defense, tracking, guidance etc [3]. In many applications, Kalman filter is generally used to estimate the true values of state variables through its observed values. In our proposed approach, the calculated values of Kalman filter are served for the displacement data, as the part of the true value of state variables. And then we use the current state vectors to predict the next state of object.

The five basic equations of Kalman filter is shown as follows, where, X is the state matrix, k represents the iteration step, A is state-transition matrix, U controlled quantity vector, P covariance matrix, Q process noise matrix, K Kalman gain vector, Z measurement(or observation) matrix, H observation matrix, and R observation noise matrix. We define B as a zero vector since no controlled quantity is used in this paper.

$$X(k|k-1) = AX(k-1|k-1) \quad (1)$$

In equation 2, where L is noise matrix, we add L to accommodate more complicated movement.

$$P(k|k-1) = AP(k-1|k-1)A^T + LQL^T \quad (2)$$

$$X(k|k) = X(k|k-1) + K(Z(k) - HK(k|k-1)) \quad (3)$$

$$K(k) = P(k|k-1)H^T(HP(k|k-1)H^T + R)^{-1} \quad (4)$$

In equation 5, we evaluate and predict the displacement and velocity together by extending KF to two-dimension.

$$P(k|k) = (I_2 - K(k)H)P(k|k-1) \quad (5)$$

Finally, input the displacement and velocity of the object in the previous frame to H, and we can obtain the current location information from X.

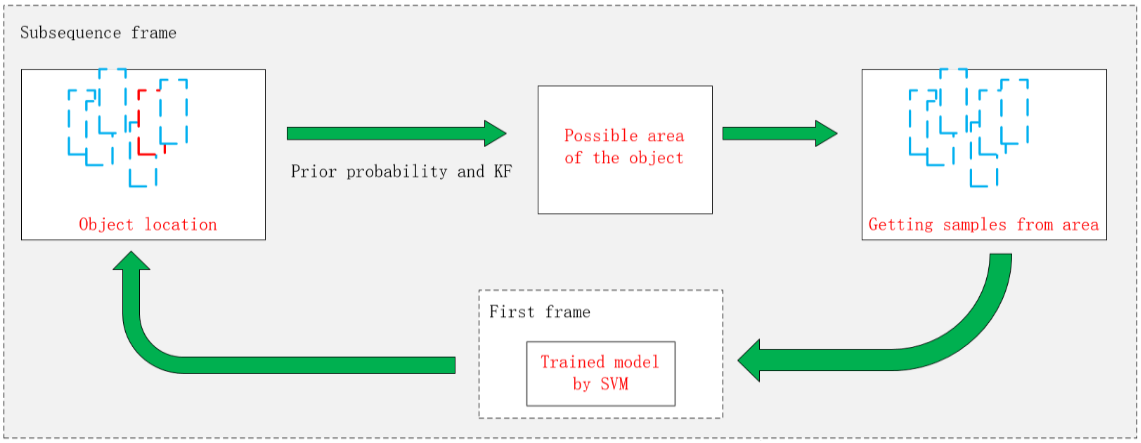

Figure 1. Overview of our tracking algorithm

2.2.2 Prior probability

Furthermore, we analyze about twenty thousand forecast errors of KF when it is given different forecast ranges over all videos in the benchmark. Consequently, those forecast error data show an attractive property, which follow the normal distribution with $\sigma \approx 1.78$, $\mu \approx 0$. It can be described by the distribution function of the normal distribution as equation 6,

$$F(n) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{n} e^{-\frac{(n-\mu)^2}{2\sigma^2}} dn = \Phi(\frac{n-\mu}{\sigma}) \tag{6}$$

where n is defined as forecast error range.

$$P(-6 \le n \le 6) = \Phi(\frac{6-0}{1.78}) - \Phi(\frac{-6-0}{1.78}) = 0.999 \tag{7}$$

In equation 7, the range of n is from -6 to 6.

$$P(-4 \le n \le 4) = 0.9752 \tag{8}$$

According to the above result, the probability of forecast error of KF is at 99.9% level within the interval {(n,o)| -6<n<6,-6<o<6}. While the probability of forecast error is at 97.5% level within the interval {(n,o)| -4<n<4,-4<o<4}. The tracking efficiency is pretty attractive even if the forecast error ranges of x and y are from -4 to 4. In summary, we can find the object of the next frame in the image patch of 8*8 centered on the KF prediction point.

The forecast error of KF algorithm mostly ascribes to two factors. On the one hand, when we use KF algorithm to predict the object motion state from the state matrix by putting the position of the previous frame into measurement matrix, the state vector is three-dimensional displacement, velocity, and acceleration information. However, the observation vector of KF is one-dimensional displacement information, and thus the forecast error is inevitable even for tracking a simple object motion. On the other hand, the actual objects move around in space. When they become two-dimensional plane motion in a video the information of one dimension will be incomplete. Definitely, the information loss of this dimension will affect the prediction accuracy of KF.

2.3 Classification Model

2.3.1 Feature vector modeling

Since the computational complexity of a matrix increases rapidly with increasing dimensions, and the feature is the key to classification accuracy, it is necessary to use an algorithm to reduce the dimensionality of the object image matrix, which is also called the feature extraction.

The information of an image shape can be described by direction density distribution of Histogram of Oriented Gradients (HOG) which is a feature descriptor based on local statistics. In this paper, as with KCF [10], we also use a HOG algorithm from Piotr's Computer Vision Matlab toolbox to extract feature vectors, where HOG features can be used to distinguish between different objects by SVM model.

2.3.2 Support vector machines

SVM is usually used for classification, and its task is to learn a classification function by training a set of samples where the binary labels can be masked as ±1 [11]. In the case of nonlinear separability situation, a mapping function (kernel function) maps the low-dimensional input vector into the high-dimensional feature vector, then the inseparable problems in low-dimensional space can be solved in high dimension.

Typically, there are four kernel functions: liner kernel, polynomial kernel, radial basis function kernel (RBF) and sigmoid kernel. In this paper, the RBF kernel function of the SVM model is widely used because of its wide convergence domain.

In the proposed approach, the SVM model solves this binary-class problem by training the classifier where the object is used as positive sample and the surrounding environments are used as negative samples. And the process of training samples is completed in the first frame. The object can be tracked by solving a kernel function.

3. EXPERIMENTS

In this section, we have used the videos from the standard benchmark dataset to test the proposed algorithm, which are challenging for tracking algorithms due to illumination changes, background clutter, pose change, fast motion and occlusion, etc.

The case experiments from the benchmark are employed to indicate the classifying performance as follows. In this paper, the code of SVM classifier is implemented by LIBSVM that is proposed by professor Chih-jen Lin at National Taiwan University. As the number of the positive sample and the negative samples defined by our approach does not match, we set the penalty parameter value: 1 and 0.125 to balance the weight of the positive and negative samples. One of the other things, We take 9 images as the training set, where these images are the positive and negative sample partition which contains the central positive sample (the object) and the surrounding eight negative samples. The visual result reveals the parameter value and sample partition are right and effective.

For evaluating the performance of the proposed method, the test result is compared against those of state-of-the-art object tracking algorithms: Struck [10], MIL [9], TLD [12], CT [8] and KCF [7]. All the experiments are executed in a PC with Intel i3 2.4GHz CPU and 4GB of RAM.

3.1 Qualitative Evaluation

We have tested all 50 videos, and four representative tracking results of the evaluated trackers are shown in Fig.2-5. The carDark sequence in Fig.2 represents low contrast between the object and background, the crossing sequence in Fig.3 represents different lighting conditions, the football sequence in Fig.4 represents background clutter and similar target obstruction, and the freeman1 sequence in Fig.5 represents pose changes.

It is obvious by checking and comparing the result images that the proposed method is effective and efficient. In carDark sequence, TLD, MIL and CT drift from the object quickly due to low contrast. In contrast, our tracker and Struck successfully track the object almost throughout the sequence. Depending on the accurately estimated image patch, the SVM model of our tracker estimates interference from the surrounding environment, while other trackers are more susceptible to the surrounding environment. In crossing sequence, Struck and TLD both track the wrong object due to background clutter. This tracking scene is relatively simple, and the background does not have much influence on the object tracking, so the trackers here, except Struck and TLD, are doing quite well. In football sequence, the object is similar to other objects in the scene, and all 6 algorithms fail to track the object at the end, while CT fails to track the object almost at the beginning. In this complex scene, there are two factors which affect the appearance model of our tracker. The first is that there are many very similar football players crowed together in the scene, which makes it difficult for SVM model to separate them, and the entire object tracking fails if there is a classification error in one frame. The second is that the appearance of the object is deformed relative to the stating frame greatly, which also has an impact on the appearance model. In freeman1 sequence, the proposed method performs well while CT and KCF gradually lose tracking the object when the face turns. In fact, our tracker also encountered the same problem, which is the change of face affects the accuracy of model. But due to the accurate estimation of image patch, our tracker can still track the object well.

The above four typical experimental results tentatively demonstrate that our proposed approach has good effect in coping with some object tracking problem caused by low contrast, different lighting conditions, background clutter and pose changes.

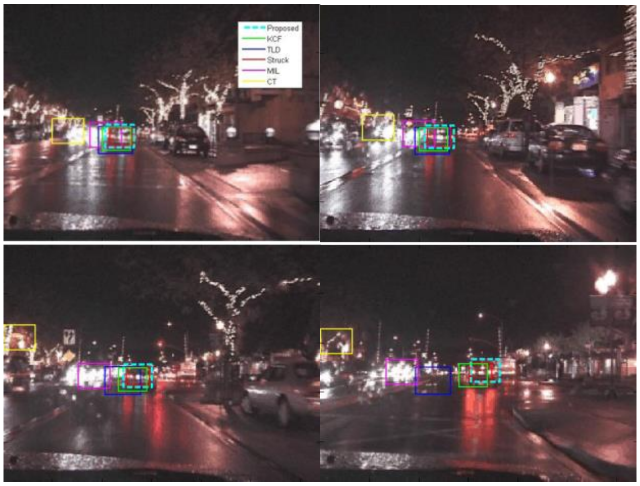

Figure 2. carDark

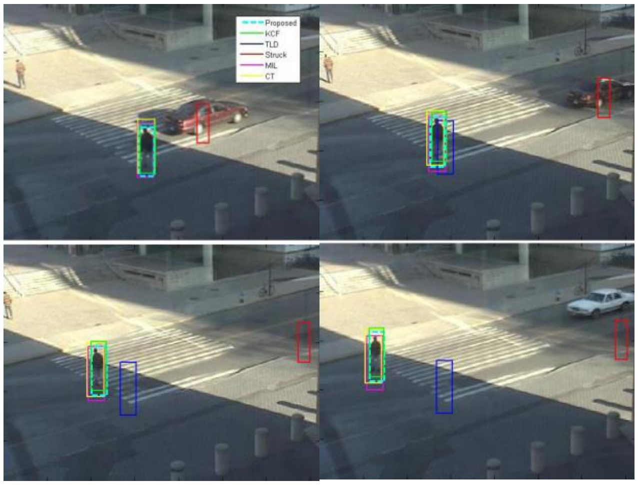

Figure 3. crossing

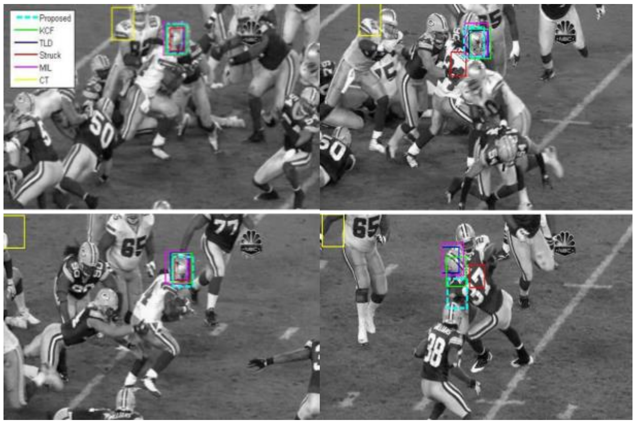

Figure 4. football

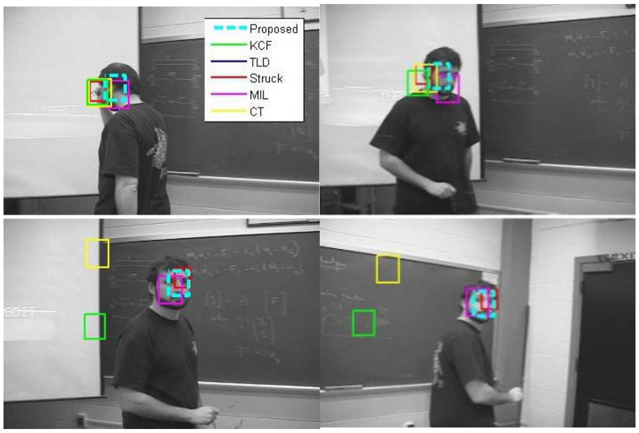

Figure 5. freeman1

3.2 Quantitative Evaluation

The global performance of the trackers is generally demonstrated by precision curves that reveal an overall performance of each method at every threshold value [13,14]. Meanwhile, we use an index of center location error metrics, for quantitative evaluation, which is defined as the average Euclidean distance between the center locations of the tracked objects and the labeled ground truths. A higher precision score at low center error thresholds means a tracker is more accurate. In addition, in this paper, we use another important evaluation indicator for trackers performance FPS because the higher FPS means the higher computational efficiency.

We test 50 videos, owing to the space reason, where 6 typical test results are shown in Fig.6-11 which depicts the precision curves that are produced by six sequences (the test results of MIL, Struck, TLD and CT come from [14]). It is considered to correctly track the object if the predicted object center is in a distance threshold of ground truth. Precision curves simply show the percentage of frames that are tracked for a range of distance thresholds correctly [9]. Comparing six figures, the proposed tracker is the robust to three of the four challenges, except for background clutter in Fig.8 that affects equally all trackers.

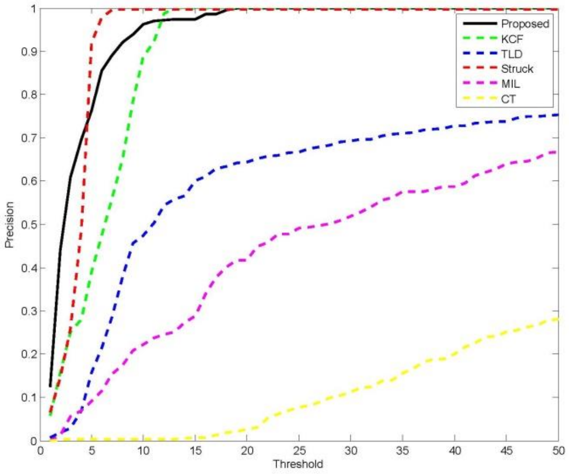

Figure 6. carDark

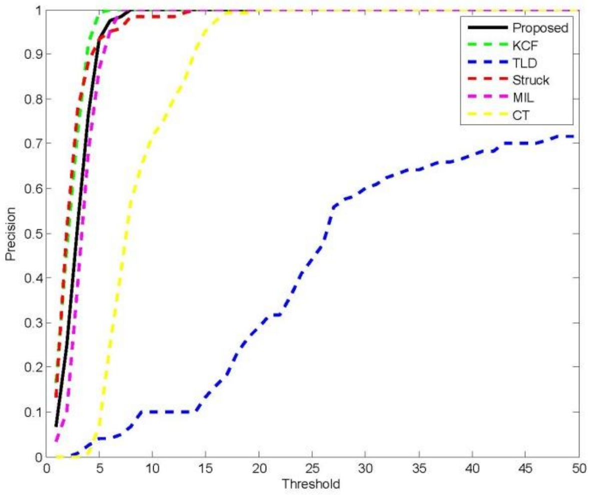

Figure 7. crossing

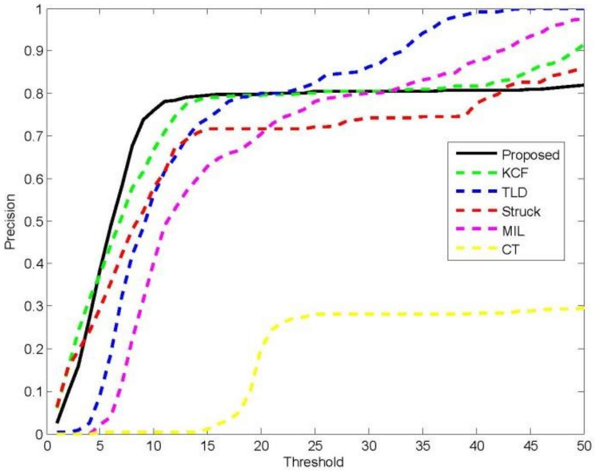

Figure 8. football

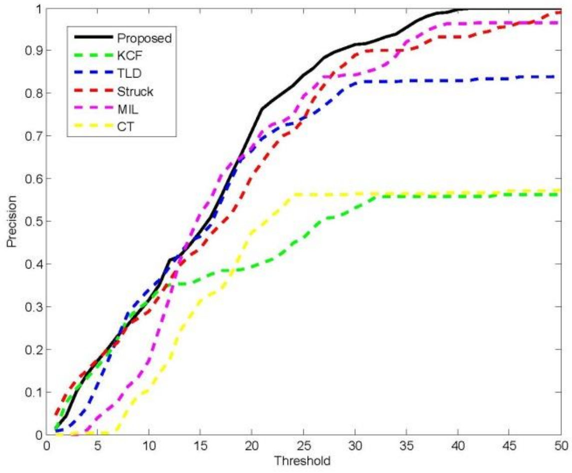

Figure 9. freeman1

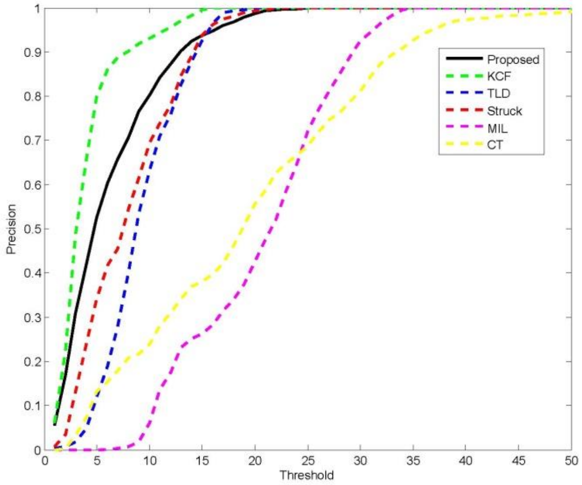

Figure 10. mhyang

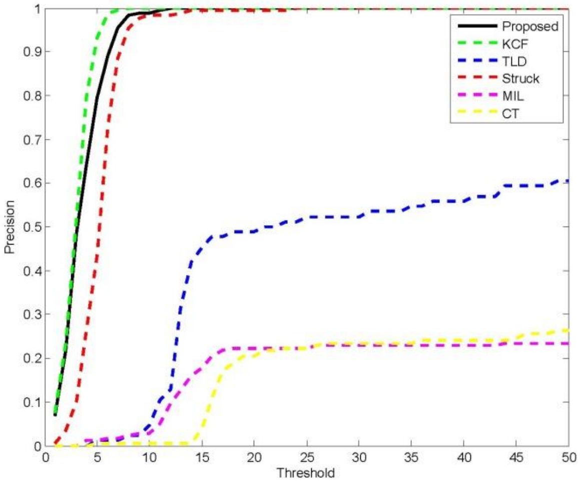

Figure 11. subway

In Table 1, we list the average frames rates [14] obtained from all 50 videos. It can be observed from the test data that the proposed method is much faster than the compared methods due to the fusion information between KF and prior probability. Our tracker is able to track arbitrary object in beyond real-time due to two aspects: First, we learn our appearance model offline, where the model only is trained in the first frame and is not updated in the subsequence frames, no online training is required. Online training trends to be slow that will prevent real-time performance of object tracking. Second, most trackers need to evaluate a large number of samples and select the one with the highest score as the tracking output. Although increasing the number of samples will improve the tracking accuracy, it will also increase computational complexity together. And our tracker estimates directly a small image patch, and gets a small number of samples with high confidence, which reduces unnecessary computational cost of object tracking, making it to complete tracking task at 200+ fps on devices without GPU.

Table 1. Average frame rates (FPS)

Tracker	FPS
Proposed	243.2
Struck [10]	11.4
TLD [12]	14.2
KCF [7]	133.3
MIL [9]	18.1
CT [8]	58.3

3.3 Algorithm Evaluation

Since the Struck algorithm loses the data of 6 videos in benchmark, our analysis just focuses on 44 test videos. The proposed algorithm tracks the selected objects throughout 22 videos successfully, and fails to track on other 22 videos that include losing or missing the objects. In the defeated videos, the objects in 20 of them are significantly deformed, and the objects in the other 2 videos are rotated. For other algorithms, KCF successes in 30 videos, TLD does in 20 videos, Struck does in 21 videos, MIL does in 7 videos, and CT does in 5 videos.

Our algorithm tries to see the tracking problem as classification problem rather than regression problem following most tracking algorithms. In the general case, this trick can always find the most similar object with the marked object. However, the problem with this trick is that our algorithm tends to make mistakes when the object has deformed or rotate because the object becomes dissimilar with the previous object after deforming or rotating.

The advantage of our algorithm is to reduce the number of samples by decreasing the image patch and speed-up the FPS of tracking, even though the tracking precision of our algorithm is not enough to compare with the KCF.

4. CONCLUSIONS

In this paper, we presented an approach for tracking arbitrary object based on a framework of motion prediction model and SVM appearance model. The proposed algorithm is divided into two steps: firstly, we introduced to use the fusion information of KF and prior probability to predict the image patch containing object at the next frame, and we obtained samples from this image patch based on predicted position. Secondly, we distinguished the object by predicting the sample with the highest score using the SVM model. From a statistical point of view, if the samples were searched in the image patch where the probability of containing the target was 97.5%, we could assure the best performance in the tracking precision and FPS. On the standard datasets, we demonstrated experimentally that the proposed computationally efficient method can outperform 5 state-of-the-art trackers against such challenges as illumination changes, pose variations, background clutter, low foreground-background contrast, partial occlusions, fast motion, etc. And the speed of object tracking is gratifying to over 200 fps.

5. ACKNOWLEDGMENTS

This work is supported by the NSFC project (No.61403356, No.61573311) and ZJNSFC project (No.LY18F030012).

6. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” Acm computing surveys (CSUR), vol. 38, no. 4, p. 13, 2006.

[2] H. Yang, L. Shao, F. Zheng, L.Wang, and Z. Song, “Recent advances and trends in visual tracking: A review,” Neurocomputing, vol. 74, no. 18, pp. 3823–3831, 2011.

[3] Sa-Ing V, Thongvigitmanee S S, Wilasrusmee C, et al. Object tracking for laparoscopic surgery using the adaptive mean-shift kalman algorithm[J]. International Journal of Machine Learning and Computing, 2011, 1(5): 441.

[4] Singh A, Kumar D, Choubey A, et al. Annotation Supported Contour Based Object Tracking With Frame Based Error Analysis[J]. International Journal of Machine Learning and Computing, 2012, 2(4): 526.

[5] Yeboah Y, Yu Z, Wu W. Robust and Persistent Visual Tracking-by-Detection for Robotic Vision Systems[J]. International Journal of Machine Learning and Computing, 2016, 6(3): 196.

[6] AlQahtani F, Banks J, Chandran V, et al. Detection and tracking of faces in 3D using a stereo camera arrangements[J]. International Journal of Machine Learning and Computing, 2019, 9(1): 35-43.

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583–596, 2015.

[8] K. Zhang, L. Zhang, and M.-H. Yang, “Real-time compressive tracking,” in European Conference on Computer Vision, pp. 864–877. Springer, 2012.

[9] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 983–990. IEEE, 2009.

[10] S. Hare, A. Saffari, and P. H. Torr, “Struck: Structured output tracking with kernels,” in 2011 International Conference on Computer Vision, pp. 263–270. IEEE, 2011.

[11] K. Ratnayake and M. A. Amer, “Object tracking with adaptive motion modeling of particle filter and support vector machines,” in Image Processing (ICIP), 2015 IEEE International Conference on, pp. 1140–1144. IEEE, 2015.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 7, pp. 1409–1422, 2012.

[13] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2411–2418, 2013.

[14] R. Collins, X. Zhou, and S. K. Teh, “An open source tracking testbed and evaluation web site,” in IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, vol. 35, 2005.