

Video Tracking Algorithm Based on Kalman Filter and Online Random Forest

Lijun Xue¹ and Lili Wang²

¹*Informatization Center, Jiangsu University, Zhenjiang, Jiangsu, 212013, China*

²*School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu, 212013, China*

¹*ljxue@ujs.edu.cn, ²llwang@ujs.edu.cn*

Abstract

Kalman filter algorithm can use target information of the previous moment to predict target information of the next moment in video tracking, but the predicted value is bound to have error. When the target movement is too fast or the target is blocked, the algorithm will converge to a region in the background that is more similar to the target color. This will cause the failure of tracking, and the tracking algorithm is difficult to accurately track the target again after tracking failure. In view of the problem of Kalman filter algorithm, this paper introduces the random forest learning algorithm in the process of Kalman filter tracking. Kalman filter output of the current frame, that is, the estimated target state, is passed into random forest detection module. This module adopts reliable sample updating strategy and rapidly detects multiple adjacent areas of the estimated target by the decision tree. For the case of transient target loss, the two methods cooperate with each other to accurately locate the tracking target. The experimental results show that the algorithm can accurately track the target, and ensure the robustness of tracking, effectively avoid the drift problem.

Keywords: *Kalman Filter, Random Forest, pHash Operator, Decision Tree, State Estimation, Confidence Degree*

1. Introduction

Video tracking technology is an important research direction in the field of computer vision, which combines computer graphics, pattern recognition, information science, automatic control and artificial intelligence technology, *etc.*, which is currently widely used in intelligent transportation, security surveillance, human-computer interaction, behavior analysis and other fields. Rudolph Kalman in 1960 broke through the limitations of Wiener filter [1], and introduced state variables and state space concept. This method didn't not require to give directly the second-order characteristic or spectral density function of signal process, but the signal process is treated as the output of linear systems under the influence of white noise. On the basis of Hilbert space projection theory [2], the theory of state estimation was put forward.

Kalman filter is a kind of optimal linear recursive filtering algorithm, suitable for multivariable system and time-varying system, and is suitable for processing non-equilibrium random process [3-7]. Kalman filter is a recursive estimates. As long as the state estimation of last moment and the observation of current state can be obtained, the current state estimate can be calculated. It don't need observed and/or estimated history record, and is a pure time-domain filter, rather than low pass filter and other frequency domain filters, need to be designed in the frequency domain, and then be transferred to time domain application. At present it is widely used in the engineering application fields such as Radar and computer vision, and is also a very important topic in the control theory

and control system engineering. Along with the Linear mean square planning, Kalman filter can be used to solve the LQG (Linear-quadratic -Gaussian control) questions.

Kalman filter can only deal with the filtering problem of the linear Gauss case effectively. In filter theory, the extended Kalman filter is the most widely used method of nonlinear state estimation. Its essence is to linearize the nonlinear state transfer model or observation model, then to use Kalman prediction and updating equations to recursively solve [8-12]. However, the linearization of the nonlinear system state space model often introduce significant error in the state estimation.

In order to better deal with the nonlinear state estimation problem, we combine the Kalman filtering algorithm and the online random forest algorithm to compensate each other, so as to improve the tracking accuracy of the algorithm.

2. Algorithm Description

This paper uses Kalman filter algorithm to predict the tracking target state, and the possible position of the target at the next moment is estimated. Then this paper takes this position as the center to search target images in a certain range, uses online random forest classifier to analyze these target images, increases judgment mechanism for features and classifier updating, uses different strategies to meet the target location of different confidence threshold, can effectively avoid the object occlusion or cumulative error caused by the weak classifier updating process.

2.1. Target Tracking Framework

The target tracking frame used in this paper is shown in Figure 1.

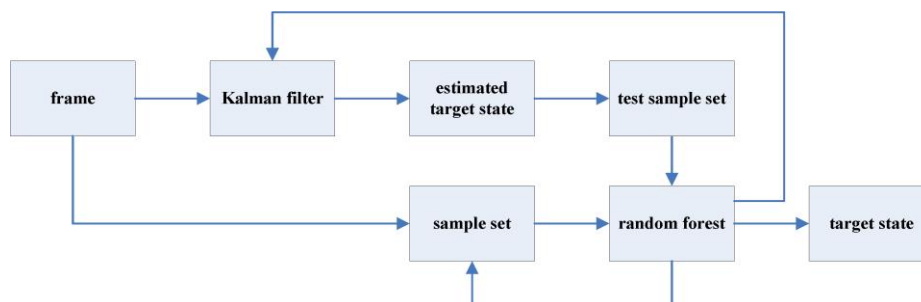


Figure 1. Target Tracking Framework

First by the Kalman filter algorithm to predict the estimate state of the target, then to take the estimated state as the center, and to obtain the neighborhood similarity target as the test sample set. The test sample set is input random forest module, and the random forest module votes for the ultimate target state. At the same time, to update predictive parameters of Kalman filter algorithm and random forest sample set.

2.2. Kalman Filter Algorithm

Kalman filter can track the moving target, but the position, velocity and acceleration of the target are always in the presence of noise. Kalman filter uses the dynamic information of the target to try to get rid of the influence of noise and get a good estimate of the target location.

To set the prediction process noise $w(n) \sim N(0, Q)$, the measurement noise $v(n) \sim N(0, R)$. Kalman calculation output is divided into the predicting process and correction process as follows:

- Step 1: prediction
- predicted value:

$$\bar{X}(n|n-1) = A\bar{X}(n-1|n-1) + Bu(n) \quad (1)$$

Minimum mean square error matrix:

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

Step 2: Correct

Error gain:

$$K(n) = P(n|n-1)H^T(n)[R(n) + H(n)P(n|n-1)H^T(n)]^{-1} \quad (3)$$

Correction value:

$$\bar{X}(n|n) = A\bar{X}(n|n-1) + K(n)[Z(n) - H(n)\bar{X}(n|n-1)] \quad (4)$$

Minimum mean square error matrix:

$$P(n|n) = [I - K(n)H(n)]P(n|n-1) \quad (5)$$

(1)~(5)中:

X(n): state vector of Nx1.

Z(n): observation vector of Mx1, input of Kalman filter.

X(n|n-1): N moment estimation results predicted by data before N moment.

X(n|n): N moment estimation results predicted by data of N moment and before N moment, this is also the output of the Kalman filter.

P(n|n-1): NxN, Minimum prediction mean square error matrix, its defined formula is as follows:

$$P(n|n-1) = E\{[X(n) - \bar{X}(n|n-1)][X(n) - \bar{X}(n|n-1)]^T\} \quad (6)$$

Finally the equation (2) is obtained by calculating.

P(n|n):NxN, the minimum mean square error matrix after correction.

K(n):NxM, error gain, from the expression of the gain, equivalent to "forecast the minimum mean squared error" divided by "measurement error of n moment + forecast least mean square error", the direct implication is the proportion of the forecast minimum mean square error of N moment state predicted by n-1 moment in total error at N moment, the greater proportion indicates the probability that truth value close to the predicted is smaller(the probability of approaching measured values is greater), this can also be seen in equation (4).

2.3. Feature Selection

pHash operator is used to describe the overall features of the image, and has the advantages of luminance invariance, color invariance and fast calculation *etc.* In this paper, pHash operator is used for feature selection.

Working process is as follows:

Step 1: reducing size: pHash begins with a small picture, but the picture is more than 8 * 8, with 32 * 32 for the best.

Step 2: converting color: converting images into gray images, simplifying the calculation quantity.

Step 3: calculating DCT: DCT transform of the image is calculated, and 32*32 DCT coefficient matrix is obtained.

Step 4: reducing DCT: the result of DCT is 32*32 matrix, but only retains 8*8 matrix of the upper left corner, which presents the lowest frequency in the picture.

Step 5: calculating average value: to calculate average value of DCT.

Step 6: calculating hash value: according to the 8 * 8 DCT matrix, set the 64 bit hash value of 0 or 1, if the value of this position is greater than that of the DCT average value, the position is set to "1", if the value of the position is less than the DCT average value, and this position is set to "0". Together, they constitute a 64 bit integer, which is pHash value of the picture.

2.4. Random Forest Algorithm

Random forest algorithm has very high prediction accuracy, and it has a very good tolerance to outliers and noise, and it is not easy to appear over fitting. Random forest algorithm is a kind of natural nonlinear modeling tool, and it is one of the most popular research fields in the field of video tracking using the random forest algorithm.

Random forest is a group of class which is composed of a series of policy tree class $\{h(X, \theta_k); k=1, \dots\}$. $\{\theta_k\}$ is Independent identically distributed random variables. When inputting X , each tree outputs its own classification results. The random forest outputs the final classification results after statistical analysis.

In this paper, each node of the classification obtains a pHash feature vector which contains feature information of feature area. This node has a simple structure, high precision, fast detection speed and fast training speed, fully meets the requirements of on-line detection.

The decision tree generation algorithm is as follows:

S: training set, R: attribute collection containing continuous attribute values, C: category attributes.

If S is empty, it returns a single node, the value is Failure;

If S is composed of the records of the same category and attribute value, it returns a single node, its value is the attribute value;

For every attribute R_i of R Do

If attribute R_i is continuous attribute, then

Begin

minimum value of R_i is assigned to A_1 :

maximum value of R_i is assigned to A_m :

For j From 2 To $m - 1$ Do $A_j = A_1 + j * (A_1 - A_m)/m$;

maximum attribute information gain (R_i, S) of the

R_i point is assigned to A ;

End;

End

The attribute (D, S) which has the maximum information gain in R is assigned to D; The value of attribute D is assigned to $\{d_j\} j = 1, 2, \dots, m$; The subset of S which is composed of records corresponding to the value d_j is assigned to $\{s_j\} j = 1, 2, \dots, m$; Its root tag is D; Branches markers are d_1, d_2, \dots, d_m ; To construct respectively D_1, D_2, \dots, D_m .

When the decision tree is set up, we can build a random forest. Random forest almost does not require input preparation, can deal with binary feature, classification characteristics, digital features, and does not require any scaling processing.

Random forest training speed is very fast. One of its characteristics is that when the performance optimization happens, it happens to improve the accuracy of the model, and vice versa. Random feature subset is designed to diversify a single tree, but also an excellent performance optimization.

Although neural network boosting algorithm model in any given data set can surpass the random forest, but compared to random forest, the establishment and adjustment of the time needed for these models is often much longer, performance is very difficult to go beyond the random forest. Random forest can easily grow in parallel, and for the boosted model and the large neural network, it cannot be.

Random forest classification results are described as follows:

First of all, to use the Bootstrap sampling to extract k sample sets from the original training set. Then, k decision tree models of k sample sets is set up, respectively, to test the test sample set for k kinds of classification results, finally, according to the result of k kinds of classification of each record to vote on the final classification decision, as shown in Figure 2.

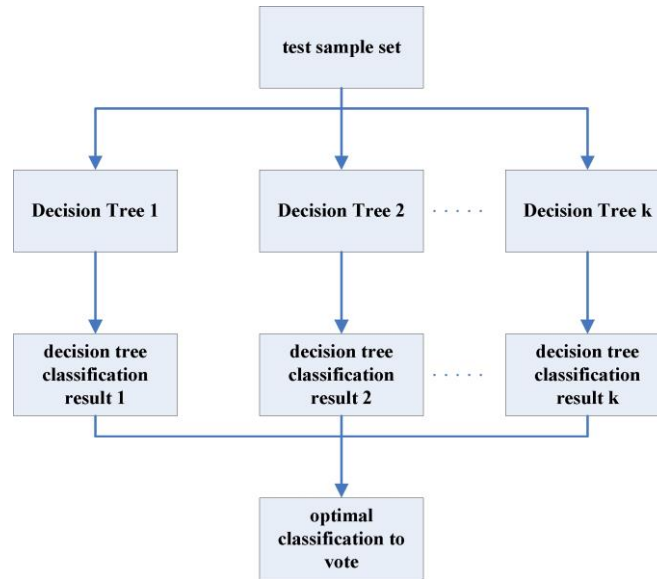


Figure 2. Decision Tree Model

A classification model sequence $\{h_1(X), h_2(X), \dots, h_k(X)\}$ is obtained by k round training, then to use them to form a multiple classification model system. The final classification decision expressed by the following formula:

$$H(x) = \arg \max_Y \sum_{i=1}^k I(h_i(x) = Y) \quad (7)$$

Among them, $H(x)$ for combination classification model, h_i for single decision tree classification model, Y for output variable, $I(\cdot)$ for indicative function.

3. The Result and Analysis of the Experiment

During the experiment, the system uses the first frame of video to initialize Kalman filter algorithm and to train random forest decision tree classifier. For subsequent frames, first, Kalman filter algorithm begins tracking targets, getting the estimated target state, then the decision tree classifier is used to detect the neighborhood of the estimated target state.

When the tracking result of Kalman filter is the same as the detection result of the voting, the other test results are false, and the ultimate target state is output, at the same time Kalman filter is updated. Kalman filter is reinitialized if the detection result of the voting and the tracking result of Kalman filter algorithm is different. Due to random forest detection module do not trace out correct tracking sample from training set, when the target is occluded or appears again, the random forest detection module can still detect the target accurately.

This experiment video comes from CCF video processing research dataset, video resolution is 360 x 288 pixels, frame rate is 25 frames per second. The input of random forest algorithm is current frame and initial tracking area box, the main function is to obtain all the scan boxes in the current frame window, the window has 21 scale, scale coefficient is 1.2, which is based on 1 intermediate scale, 10 times 1.2 times reduction, 10 times 1.2 times magnification. Under each scale, the scanning step size of the window is 10% of wide and high (SHIFT = 0.1), and get all the scan window and store in a container grid, every element of the grid contains six attributes: the top left corner coordinates x, y, width w, height h of the current scan window, and overlap rate overlap of initial tracking

area box, the size of the current scan window. Overlap rate is defined as: the intersection of two windows/two windows and set. And to extract 100 positive samples of largest confidence from video subsequent each frame to join the training set.

In the experiment, we first use Kalman filter algorithm to track the moving target of the experimental video. Frame 3, frame 41, frame 117, frame 118, frame120, frame 228 tracking results of Kalman filter algorithm are as shown in Figure 3.

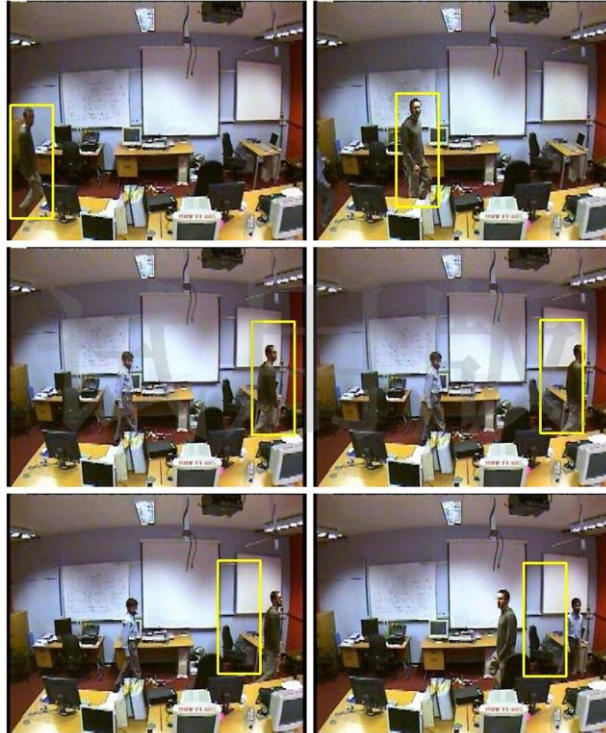


Figure 3. Tracking Results of Kalman Filter Algorithm

Secondly, we use the proposed algorithm to track the moving target in the same experimental video. Frame 3, frame 41, frame 114, frame 115, frame 116, frame 228 tracking results of our method are as shown in Figure 4.



Figure 4. Tracking Results of the Proposed Algorithm

The tracking results confidence scattergram of the proposed algorithm is as shown in Figure 5.

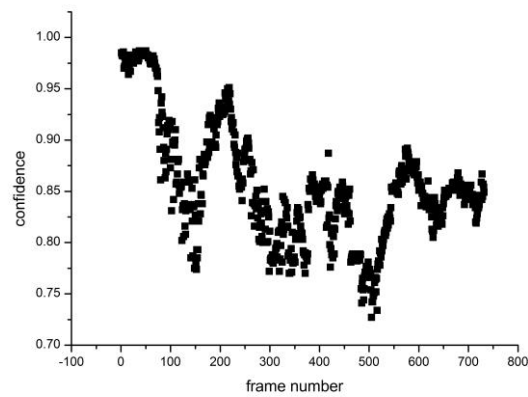


Figure 5. Tracking Results Confidence Scattergram

Through the analysis, when the confidence is greater than 0.75, the system basically uses the Kalman filter tracking algorithm, if the confidence level is less than 0.75, the system will introduce random extracting edges of search area as negative samples to sample set for further training, and form joint training effect, therefore, the increase of distractors only will temporarily affect the tracking result, target is frequently obscured, leave the picture, is perfectly normal for a drop in the confidence.

In the experiment, using Kalman filter tracking algorithm, frame 2 to 117 frames can accurately track the target. Using the method, the second frame to the end of the video can accurately track the target. But to the 138th frame, after the target running state changes,

characteristics examined by Kalman filter change, detection will always stay near the target position, target tracking lost.

In the experiment, in order to accurately quantitative comparison tracking results, to use the following formula to calculate tracking error of the k-th frame position:

$$E_k = |T_k - C_k| \quad (8)$$

T_k indicates the location of the tracking object in the k-th frame, C_k indicates the exact position of the object in the k-th frame, E_k indicates the absolute value deviations from precise location, and can be used to measure the accuracy of tracking, as the tracking error.

To use the following formula to measure the average absolute deviation of target tracking:

$$\bar{E} = \sum_{k=1}^N E_k / N \quad (9)$$

Respectively using above two formulas to calculate the x, y direction absolute value deviation and average absolute value deviation of target location in each frame. Among them, C_k is accurate target center position of each frame in the test sequence obtained by calibration method in advance.

Figure 6 and Figure 7 shows respectively the absolute value deviation of the tracking results in x direction and y direction based on Kalman filter and online random forest video tracking algorithm proposed in this paper and the traditional Kalman filter tracking algorithm.

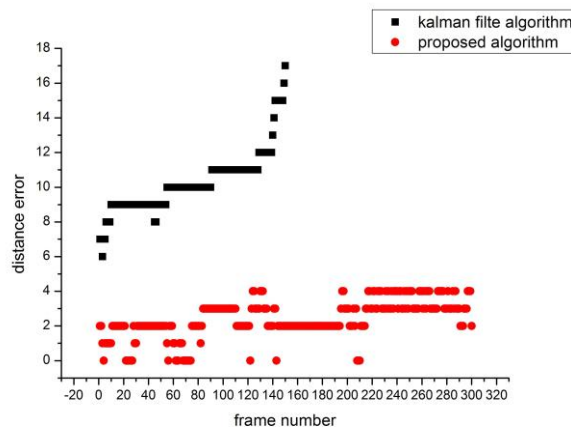


Figure 6. The Tracking Error of Absolute Value in X Coordinates Comparison Diagram Based on Kalman Filter Algorithm and Proposed Algorithm

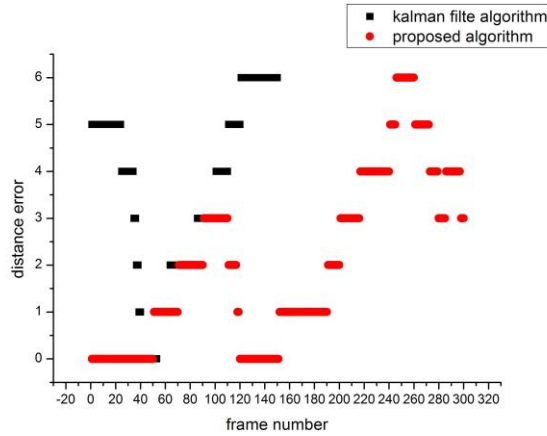


Figure 7. The Tracking Error of Absolute Value in Y Coordinates Comparison Diagram Based on Kalman Filter Algorithm and Proposed Algorithm

The average absolute deviation of track results based proposed algorithm and Kalman filter tracking algorithm is shown in Table 1.

Table 1. Average Absolute Deviation Compared

algorithm	X direction \bar{E} (pixel)	Y direction \bar{E} (pixel)
Kalman filter algorithm	10.3	3.6
proposed algorithm	2.4	2.1

According to Figure 6 and Figure 7 and Table 1, quantitative performance comparison of target tracking based on Kalman filter algorithm and proposed algorithm can be more intuitive, due to the scale and speed of the target change constantly in the process of tracking, although the traditional Kalman filter tracking algorithm able to keep up with the target in the start phase of tracking, but the tracking results aren't stable enough. In the second half of the tracking sequence, the target scale and velocity changes in a rapidly changing, the absolute value deviation of tracking results based on Kalman filter tracking algorithm increases sharply so that the tracking performance significantly decreased.

Although the proposed algorithm will be affected by target dimension and speed change, but the absolute deviation of tracking results is less than the results of Kalman filter tracking algorithm, embodying the improvement of the algorithm tracking performance. In the proposed algorithm, pHash operator is used in the random forest, and is joined the classifier, make use of previous video frame to train classifier. When updating classifier, lower confidence training samples will be not join in the sample library. Thus, this algorithm can well improve Kalman filter tracking algorithm, and has better adaptability and robustness to the target state change.

4. Conclusions

In this paper, Kalman filter algorithm and random forest algorithm are discussed, and the tracking of moving target in complex background is studied. The moving target tracking in video sequences is realized by the combination of Kalman filter algorithm and random forest learning algorithm. A rough estimate of the target state is given by using Kalman filter computing framework, and the estimated target position is used as the center, and the samples are collected in a small range as the test samples. After object

occlusion or missing, the decision tree detection module rapidly detects test samples of target state set, and uses a reliable sample updating strategy. The two methods cooperate with each other finally accurately locate the target tracking, to ensure the robustness of tracking, effectively avoid the drift problem.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (No. 41374129).

References

- [1] Z.-L. Deng, "Kalman filtering and Wiener filtering ---- modern time series analysis method", Harbin Institute of Technology press, Harbin, (2001).
- [2] R. E. Kalman, "A new approach to linear filtering and prediction problems", Transactions of the ASME, Series D, Journal of Basic Engineering, vol. 82, (1960), pp. 34-45.
- [3] Yuan Tianxin, "The best estimate principle", National Defense Industry Press, Beijing, (1980).
- [4] T. G. Robertazzi and S. C. Schwartz, "On applying the extended Kalman filter to nonlinear regression models", IEEE Transactions on Aerospace and Electronic Systems, vol. 25, no. 3, (1989), pp. 433-438.
- [5] G. A. Einicke and L. B. White, "Robust extended Kalman filtering", IEEE Transaction on Signal Processing, vol. 47, no. 9, (1999), pp. 2596-2599.
- [6] H. Grabner and H. Bischof, "On-line Boosting and Vision", Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2006); New York, USA.
- [7] Z. Kala, J. Matas and K. Mikolajczyk, "Tracking-Learning-Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, (2012), pp. 1049-1422.
- [8] E. Maggio and A. Cavallaro, "Video Tracking: Theory and Practice", John Wiley and Sons, (2011), pp. 45-59.
- [9] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, (2003), pp. 564-577.
- [10] A. D. Jepson, D. J. Fleet and T. F. El-Maraghi, "Robust online appearance models for visual tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, (2003), pp. 415-522.
- [11] S. Avidan, "Support vector tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, (2004), pp. 1064-1072.
- [12] P. Tissainayagam and D. Suter, "Object tracking in image sequences using point features", Pattern Recognition, vol. 38, no. 1, (2005), pp. 105-113.

Authors



Lijun Xue, is studying in Jiangsu University for the Ph.D. degree. His main research interests include computer vision processing and computer network.



Lili Wang, is a computer teacher in Jiangsu University. Her main research interests include computer vision processing and digital image processing.