International Journal of Recent Engineering Science (IJRES) Volume 6 Issue 5, 1-6, Sep-Oct 2019
ISSN: 2349 – 7157 / doi: 10.1445/23497157/IJRES-V6I5P101

Target detection and tracking system based on IMM

Yizhao Wang[1], Shuner Chen[1], Weiping Liu[1] *
[1]*Department of Electronic Engineering, College of Information Science and Technology, Jinan University, Guangzhou, 510632, China*

Abstract

At present, video target tracking mainly uses correlation filters and deep learning. In target tracking, deep learning often uses convolutional neural networks and has strong feature extraction capabilities. However, due to the large amount of computation required for deep learning, the GPU performance requirements are high, and the computation time is extended, resulting in unsatisfactory tracking speed. Therefore, it is common to apply correlation filters for processing. The more classical ones are the nucleation correlation filter KCF and the Kalman filter. Both processing speeds are significantly improved compared to deep learning.

In comparison, the Kalman filter is more advantageous because the moving position of the adjacent frame target is larger or more extensive. Simulate the motion trajectory of the target by the given data, introduce noise as the observation trajectory, and then analyze the error before and after the filtering by Kalman filtering; fit the complex trajectory. The video target tracking program of the interactive multi-model Kalman filter is obtained, and the target tracking accuracy before and after the application of IMM improvement is analyzed.

Keywords*: Video target tracking; Kalman filtering; interactive multi-model IMM*

I. INTRODUCTION

About 80% of the information received by humans through the senses comes from vision. Using related equipment such as computers and video cameras, the simulation of biological vision can be realized. This is called computer vision. By processing the acquired picture or video to obtain three-dimensional information of the corresponding scene[1], the two-dimensional projection of the three-dimensional world is transmitted to the computer, and the computer system can process the visual information.

The main task of moving target tracking is to design methods and models to detect moving objects in a continuous image or video. The main problems are real-time, robustness, and accuracy. Among them, due to the uncertainty of the motion of the object and the mutual obscuration of the moving target, the target information is easily lost, and the Tracking fails. The leading solutions are as follows: using multiple camera construction systems to deal with multiple moving target mutual occlusion problems, using the EM algorithm[2] to establish Tracking based on color information on the tracked human body, region-based Tracking, model-based Tracking, activity-based Tracking of contours. The main drawbacks of these algorithms are that the algorithm is more complex and cannot meet the requirements of real-time performance. Under the existing PC hardware conditions, the requirements for real-time detection and Tracking of moving targets cannot be achieved.

Besides, video target tracking also needs to be able to adapt to video noise caused by picture light and shadow changes, partial occlusion, complex background, camera movement, target object size changes to ensure accurate Tracking;

On the other hand, real-time online Tracking must be performed between image frames without affecting the playback speed of the video. Otherwise, the information is poorly conveyed, or the tracking accuracy impaired. A qualified tracking algorithm must have sufficient stability and robustness, and it needs to be able to run at high speed in real-time to prevent tracking delay or failure.

This paper uses MATLAB to simulate Kalman filter and IMM Kalman filter algorithm[3], [4], compares the performance of Kalman and IMM Kalman, uses Kalman filter to improve the accuracy of video target tracking, and optimizes video tracking algorithm by using interactive multi-model to achieve high-precision uninterrupted Tracking.

II. HEADINGS

First, we use Patel H A 's[6] classic paper "Moving object tracking using Kalman filter" to understand the processing flow of video target tracking and the role of Kalman filter in target tracking:

1) Background extraction: We regard the background as a signal, and the motion target as

noise, i.e., y(i)=x+v(i), where y(i) denotes the signal received by measurement, and x denotes the background The signal, v(i), represents noise (target). When the background is unchanged, according to the Gauss-Markov theorem, averaging all the noisy signals, that is, the background signal.

2) Noise (target) extraction: For each frame, subtract the background extracted from the first step to get the target to be tracked. For true noise processing, since the target area is generally larger than noise, a simple method of ignoring a small area is used to filter out the influence of noise. The above two steps are the process of target detection.

3) Perform Kalman filtering, divided into two steps of prediction and update, and obtain the filtered estimation value.

4) Finally, the article tested three different situations: insufficient sample size, large noise, and Kalman filter input (observation) interruption. The experimental results are as follows: green represents the small sphere entity, black represents the predicted sphere position, and yellow represents the estimated position after Kalman filtering. The closer the small yellow circle to the green ball is, the better the filter tracking effect is.

III. INDENTATIONS AND EQUATIONS

The IMM uses multiple Kalman filters[7] for parallel processing. Each filter describes different motion modes, such as uniform linear motion, left turn, right turn. At each moment, the initial condition of a particular filter the state estimation value obtains this moment at the last moment and the corresponding state probability, and then each model is predicted and corrected in parallel. Then the model is updated by the model matching likelihood function. Finally, based on the model probability, the final state estimate is obtained from the weighted sum of the state estimates of all the filters.

Assuming that the target has r motion states, the IMM can contain r models, i.e., r state transition equations.

Then the target state equation of the jth model is:

$$X_j(k+1) = \Phi_j(k) X_j(k) + G_j(k) W_j(k) \quad (1)$$

The observation equation is:

$$Z(k) = H(k) X(k) + V(k) \quad (2)$$

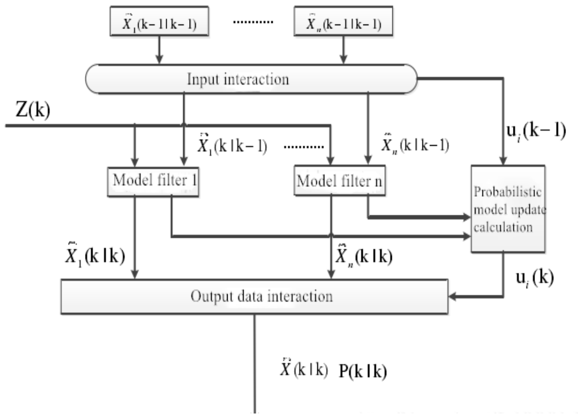

Fig1. IMM algorithm flow chart. The figure mainly describes the interaction of the output data of the IMM algorithm input data after the model filter is mentioned, and the status update occurs.

Among them Q_j is a white noise sequence with a mean $W_j(k)$ of 0 and a covariance. The transition between the models is determined by the Markov probability transfer matrix, which p_{ij} represents the probability that the i-th model is transferred to the j-th model. The model transition probability is expressed as follows:

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1r} \\ \vdots & \ddots & \vdots \\ p_{r1} & \cdots & p_{rr} \end{bmatrix} \quad (3)$$

The IMM algorithm is divided into four stages for recursion: input interaction, Kalman filtering of model j, model probability update, and output interaction.

1) Input interaction of model j.

The mixed state estimate $\hat{X}_{0j}(k-1|k-1)$ and the covariance $P_{0j}(k-1|k-1)$ are obtained from the state estimation $\hat{X}_i(k-1|k-1)$ of the target and the model probability $\mu_j(k-1)$ of the filter in the previous step, and the hybrid estimation is taken as the initial state of the current cycle. The predicted probability (normalized constant) of model j is:

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij}\mu_i(k-1) \quad (4)$$

The mixing probability of model i to model j is:

$$\mu_{ij}(k-1|k-1) = \sum_{i=1}^{r} p_{ij}\mu_i(k-1)/\bar{c}_j \quad (5)$$

The mixed state of model j is estimated as:

$$\hat{X}_{0j}(k-1|k-1) = \sum_{i=1}^{r}\hat{X}_i(K-1|K-1)\mu_{ij}(k-1|k-1) \quad (6)$$

The mixed covariance of model j is estimated as:

$$P_{0j}(k-1|k-1) = \mu_{ij}(k-1|k-1) * \{P_i(k-1|K-1) + \sum_{i-1}^{r}[\hat{X}_i(k-1|k-1) - \hat{X}_{0j}(k-1|k-1)] * [\hat{X}_i(k-1|k-1) - \hat{X}_{0j}(k-1|k-1)]^T\}$$

(7)

Where p_{ij} is the transition probability of model i to model j, and $\mu_i(k-1)$ is the probability of model i at time k-1.

2) Kalman filtering of model j.

Perform Kalman filtering with $\hat{X}_{0j}(k-1|k-1)$, $P_{0j}(k-1|k-1)$ and Z(k) as the initial inputs of model j, update prediction probability $\hat{X}_j(k|k)$, and filter covariance $P_j(k|k)$.

State prediction:

$$\hat{X}_j(k|k-1) = \Phi_j(k-1)\hat{X}_{0j}(k-1|k-1) \quad (8)$$

Error covariance prediction:

$$P_j(k|k-1) = \Phi_j P_{0j}(k-1|k-1)\Phi_j^T + G_j Q_j G_j^T \quad (9)$$

Kalman coefficient:

$$K_j(k) = P_j(k|k-1)H^T[HP_j(k|k-1)H^T + R]^{-1} \quad (10)$$

Status update:

$$\hat{X}_j(k|k) = \hat{X}_j(k|k-1) + K_j(k)[Z(k) - H(k)X_j(k|k-1)] \quad (11)$$

Covariance matrix update:

$$P_j(k|k) = [I - K_j(k)H(k)]P_j(k|k-1) \quad (12)$$

3) Model probability update

The likelihood function of model j is:

$$L_j(k) = \frac{1}{(2\pi)^{n/2}|S_j(k)|^{1/2}}\exp\{-\frac{1}{2}v_j^T S_j^{-1}(k)v_j\} \quad (13)$$

$$v_j(k) = Z(k) - H(k)\hat{X}_j(k|k-1) \quad (14)$$

$$S_j(k) = H(k)P_j(k|k-1)H(k)^T + R(k) \quad (15)$$

Based on this, update the model probability:

$$\mu_j(k) = \frac{L_j(k)\bar{c}_j}{c}$$

(16)

Where the $c = \sum_{j=1}^{r} L_j(k)\bar{c}_j$ is normalized constant.

4) Output interaction

According to the estimation result of each filter, and the model probability, the weighted sum of them can be obtained:

Total state estimate:

$$\hat{X}(k|k) = \sum_{j=1}^{r}\hat{X}_j(k|k)\mu_j(k) \quad (17)$$

Total covariance estimate:

$$P(k|k) = \sum_{j=1}^{r}\mu_j(k)\{P_j(k|k) + [\hat{X}_j(k|k) - \hat{X}(k|k)] * [\hat{X}_j(k|k) - \hat{X}(k|k)]^T\} \quad (18)$$

The total output of the filter is the weighted value of the multiple filter estimation results, and the weight is the model probability (representing the probability that the model correctly describes the target motion at that moment).

When choosing the motion model of the filter, the following rules 21 should be followed:

(1) When the target is in a continuous motion state, it can generally be expressed by a more accurate model. When the target state is in an abrupt state, it is often difficult to express it in single or multiple models. In this case, a rougher model is needed. The IMM filter should include models for both cases

(2) The Markov chain state transition probability greatly influences the model error, so it should be considered to match the degree of change of each state model when selecting.

(3) There should be a modular idea. When subjectively clear the target motion law, choose a more accurate model. When the target motion law cannot be predicted, a more general

model should be selected to reduce the estimation as much as possible—error 21.

Based on the above principles, this paper carries out MATLAB simulation on the application of target tracking in IMM to analyze its application feasibility:

Firstly, a set of data is generated to simulate the primary curve motion of the target, then the random noise simulation observation process is added, and finally, the Kalman filter of the interactive multi-model is realized by IMM to obtain the estimated value. The Monte Carlo method was used for 50 simulations to observe the observed trajectory and the filtered trajectory to see which one is more suitable for the real trajectory. The experimental results are as follows: (pictured as motion, observation, and estimation trajectory)

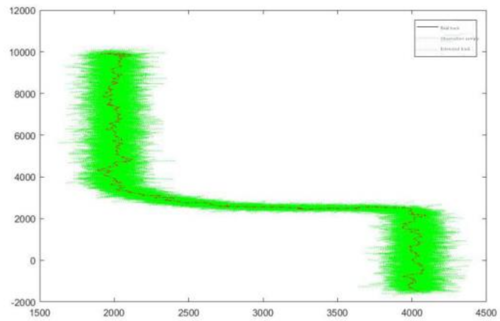

Figure 2. Motion, observation, and estimation trajectory before and after IMM Kalman filtering.

It can be seen from the result of fig. 2 that after the introduction of noise, the observed trajectory swing is enormous. In contrast, the estimated trajectory after the IMM has oscillated in a small range around the real value, which can better track the motion of the target. Get a more accurate trajectory. Therefore, the interactive Kalman filter can be better applied to target tracking.

IV. EXPERIMENT

IMM integrates multiple states, not just a single state. Therefore, compared with traditional Kalman filtering, it can quickly adapt to the state change of the target and has an excellent correcting and smoothing effect on the trajectory tracking of multi-model moving targets. The ability is much improved, and the movement of the target can be tracked more accurately. The following is a comparison of the filtering performance of IMM and Kalman through MATLAB simulation:

Unlike the fig. 2 programs, in the 450 samples, the first 20 times are estimated by Kalman filtering, and after 20 times, the IMM is used for filtering, and the filtered error standard deviation is obtained. The simulation results are as follows:

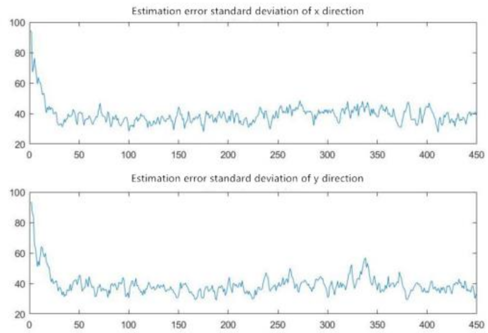

Fig. 3 Estimation error standard deviation of the x-direction

It can be seen from the results in fig. 3 that the error standard deviation of applying Kalman filtering is large when N<20, and the standard deviation of error of applying IMM Kalman filtering is smaller and stable when N>20. The error standard deviation is one of the most important indicators to measure the tracking performance. Therefore, after applying the interactive multi-modal improvement, the tracking accuracy of the Kalman filter has been dramatically improved. You can summarize the performance difference between the two:

(1) When the target motion state is uniform linear motion, the Kalman filtering result is more accurate; when the target motion state changes frequently, IMM Kalman performs better than Kalman filtering.

(2) The robustness of Kalman filtering is far less than that of IMM Kalman;

(3) IMM has a defect: the calculation amount is large, and the processing is slow. If the algorithm is implemented using a programmable device, the main frequency requirement for the hardware is high.

Target detection part:

1) realized by MATLAB function vision.PeopleDetector, after step processing, set two parameters returned: bbox and scores, where bboxes = (xy width height) represents the position and size of the rectangular frame containing the target, using This parameter can be used to mark the target; if size(scores) > 0, it means that a person is detected in this frame, otherwise it is not. Use this parameter to

find the first frame containing the target in successive frames.

2) Target tracking part: Define the target variable mu = (x y vel_x vel_y) to indicate the position and speed of the target in both x and y directions. Define the measurement variable y_t = [bboxes(1, 1) + bboxes(1, 3)/2; bboxes(1, 2) + bboxes(1, 4)/2]; represents the center position of the target.

At this point, the entire process of one frame of target tracking has been completed, and the effect interface is shown in Figure 4:

Fig. 4 Improve the previous target tracking effect chart

Different colors represent the target detection, prediction, and Kalman filtering results, in which the size of the prediction and update boxes are fixed. Intuitively, the higher the coincidence of the update box update and the center of the target, the more accurate the Tracking; from the data point of view, the size of the ellipse represents the magnitude of the error covariance as a measure of the tracking accuracy.

Defects in applying Kalman filtering in video target tracking:

(1) When the target moves in a uniform linear motion, the overall tracking effect is better, and the Update box can accurately frame the target. However, in the video, when the person crosses the road, the Kalman filter tracking result is not satisfactory in the process of target state transition:

Fig. 5 Improve the effect of the previous target state change

As can be seen visually in Figure 5, the update box deviates more from the center of gravity, and the diameter of the ellipse box representing the error covariance also becomes large; that is, the tracking accuracy is significantly reduced. It can be seen that Kalman filtering is not suitable for processing video target state transitions frequently.

(2) Even if the target in the video moves forward (away from the camera), that is, when it is roughly linear, the center of gravity is not a perfect uniform linear motion because of the slight fluctuation of the person walking. However, the tracking frame processed by Kalman filtering can not capture this point, and it is still roughly traced by the trajectory of uniform linear motion. In terms of accuracy, the processing effect of Kalman filtering is not satisfactory.

Compared with Kalman filtering, the application of interactive multi-model IMM Kalman filter in video target tracking can better adapt to the replacement of the target state. However, the target in the actual video is usually not a single uniform linear motion. Therefore, it is feasible to improve the tracking accuracy of the video target by using the Kalman filter based on the interactive multi-model IMM.

The improved overall framework for tracking video target tracking using IMM is the same as that described in fig4: target detection and labeling, IMM filtering, update status values , and error covariance. The difference with fig4 is that IMM improves the kalman algorithm, and target tracking is performed.

The IMM filtering part mainly defines three basic models: model one is non-maneuverable, and models 2 and 3 are maneuver models (Q values are different). Using the weighted sum of these three models as the estimation model, the model coefficients and weighting coefficients are updated iteratively. The algorithm principle has been introduced in fig. 3 is divided into four phases: input interaction, Kalman filtering of model j, model probability update, and output interaction. The filtering and tracking effects are also presented by the three rectangular boxes of People Detector, prediction, update, and ellipse, no different from fig. 4.

The experimental results are as follows:

(1) As shown in fig. 6 is the tracking effect diagram after the Kalman filtering is improved by I MM. It can be seen from the video that the tracking frame has a slight up and down sway with the fluctuation of the center of gravity. It shows that the

improved algorithm can capture slight changes in the center of gravity of the person, and the tracking effect of the improved Kalman filter is not so accurate. Intuitively, the tracking accuracy is much improved.

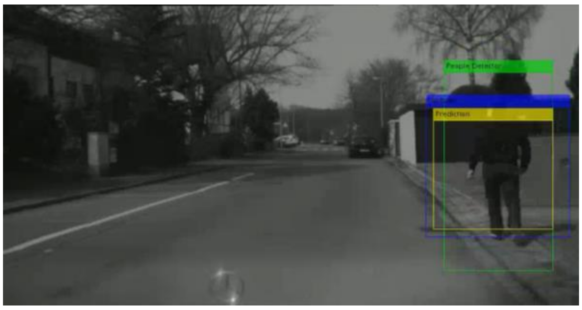

Fig. 6 Improved target tracking rendering

When the target state changes (when people cross the road), the visually improved tracking frame deviates more from the target real trajectory than before the improvement, as shown in Figures 7 and 8:

Fig. 7 Effect of improved target state change Figure 1

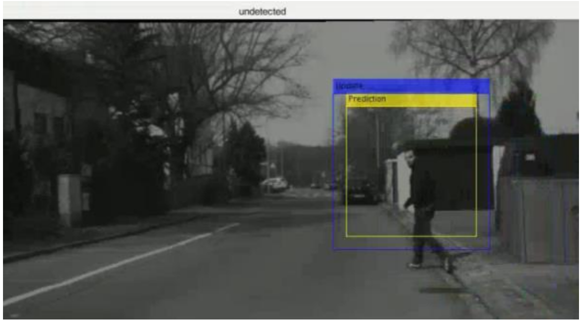

Fig. 8 Effect of improved target state change Figure 2

Compared with Figure 5 (the effect diagram when the target state changes before the improvement), it can be seen that the tracking frame deviates more from the target. The reasons for the analysis are as follows:

1) In the process of crossing the road, the target is always undetectable (see the undetected mark in Figure 8, indicating that the frame cannot detect the target; if the target is detected, it is detected), and the detection frame is never present. Therefore, from the first frame crossing the road, the filter has no input value. It is known from Section 2.4 analysis that in this case, the predicted and updated values of the filter are entirely dependent on the state estimation value and error agreement of the previous frame—an estimate of variance. Therefore, the goal is to make a uniform linear motion during the crossing, and its speed and direction depend on the speed and direction of the target when the last frame can be detected.

2) From the above analysis, the improved program is more sensitive to changes in the target state. Therefore, as long as the person has just passed the road (when the state changes) when the last frame of the person is detected, there is a slight change in the center of gravity. The IMM Kalman filter was used as the signal for the state change, so as the final input signal to correct the target tracking. Frame. Therefore, after a prolonged interruption of input, the tracking frame of the IMM Kalman filter is more likely to deviate from the actual motion trajectory of the target.

Then analyze and compare the advantages and disadvantages of Kalman filtering and IMM Kalman filtering.

V. CONCLUSION

From the above analysis of the experimental results, it can be seen that the improved tracking accuracy has been significantly improved. However, because the detector part of the MATLAB integrated detection program is not robust enough, the target is not detected in the critical process of the target state transition, resulting in improvement. The tracking frame deviates more. So there are two ways to improve:

1) Improve the robustness of the detection part

2) re-record the video to ensure that basically, every frame can identify the detected target.

REFERENCES

[1] Blackman, Samuel S. *"Multiple-target tracking with radar applications."* Dedham, MA, Artech House, Inc., 1986, 463 p. (1986).

[2] Wu, CF, Jeff. "*On the convergence properties of the EM algorithm."* The Annals of Statistics 11.1 (1983): 95-103.

[3] Liao, Jung-Feng, and Bor-Sen Chen. *"Robust mobile location estimator with NLOS mitigation using interacting multiple model algorithm."* IEEE Transactions on Wireless Communications 5.11 (2006): 3002-3006.

[4] Munir, A., and D. P. Atherton. *"Adaptive interacting multiple model algorithm for tracking a maneuvering target."* IEE Proceedings-Radar, Sonar and Navigation 142.1 (1995): 11-17.

[5] Ms. G.Anusha , Mrs. E. Golden Julie. *"Improving the Performance of Video Tracking Using SVM"*, International Journal of Engineering Trends and Technology (IJETT), V11(3),133-139 May 2014. ISSN:2231-5381.

[6] Patel H A, Thakore D G. *"Moving object tracking using Kalman filter"*[J]. International Journal of Computer Science and Mobile Computing, 2013, 2(4): 326-332.

[7] Drolet L, Michaud F, Côté J. *"Adaptable sensor fusion using multiple Kalman filters[C]//Proceedings"*. 2000 IEEE/RSJ

International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113). IEEE, 2000, 2: 1434-1439.