Sensors & Transducers, Vol. 173, Issue 6, June 2014, pp. 6-15

Sensors & Transducers

Development of Multi-target Tracking Technique Based on Background Modeling and Particle Filtering

[1, 2] Liao Xue-Chao, [3] Liu Zhen-Xing
[1] School of Computer Science, Wuhan University of Science and Technology, Wuhan 430081, P. R. China
[2] Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan 430081, P. R. China
[3] School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, P. R. China
[1] Tel.: 0086-027-68893240, fax: 0086-027-68893240
[1] E-mail: liaoxuechao2008@sina.com

Received: 24 December 2013 /Accepted: 8 May 2014 /Published: 30 June 2014

Abstract: Based on implementing target tracking by means of particle filtering, a technique framework of tracking target by integrating particle filtering and background modeling is presented. The multi-target tracking (MTT) is classified into 5 modules as background modeling, multi-target tracking, initializing, re-initializing and particle filtering. Firstly, the author models each pixel of the image with Gaussian Mixture Model (GMM) to calculate the probability of background pixel in the current image so as to abstract foreground moving objects. Based on the background modeling, the algorithm flow and technique framework of generating the particle set of each object and particle filtering are presented. In the process of evaluating particle weight, in order to distinguish the different color features of the objects, the original algorithm (evaluating through Bhattacharyya distance) is improved. Only the color distribution of the foreground pixel in particle area after the background modeling is counted, therefore the accuracy and efficiency of target tracking are increased. The experiments prove that this algorithm can realize the effective tracking several moving persons.

Keywords: Particle filter, Multi-target tracking, Gaussian mixture model, Weight evaluation.

1. Introduction

MTT in image sequence is always a very active research subject in the fields like computer vision, image processing and pattern identification etc. The process of MMT is to detect each independent moving target or area in each frame, then position these targets or area in the followed frame so as to obtain the integrated moving track of each moving target. The methods mainly adopted are moving targets detecting and targets tracking.

The purpose of moving targets detecting is to extract the changing region from background image sequence. The background modeling is a common method of detecting moving targets. Most of researchers devote themselves to developing different background models, so as to reduce the influence by dynamic scene variation on moving extracting. For

instance, Karman and Brandt [1], Kilger [2] use Kalman filtering self-adaption background model to adapt to the time diversification of weather and illumination. Thanarat Horprasert [3], by building the color model of brightness and chrominance variation, distinguishes the foreground targets from shadow background, highlight background and common background, then chooses threshold value automatically to classify pixel Picture element. Stauffer and Grimson [4] use self-adaption Gaussian Mixture background modeling [5], and utilize on-line estimate methods to update modeling. Moving targets tracking is equivalent to building the corresponding matching problem based on corresponding features (location, speed, shape, texture and color) in image sequence. The widely used mathematical tools are Kalman filtering [6], Condensation [7] algorithm, dynamic Bayesian network [8], etc.

Particle filtering, as a nonlinear filtering based on Bayesian estimation, has the advantage of tracking nonlinear moving targets. Gordon [9] etc., proposes the resampling thought in recursion progress to develop the particle filtering technique, subsequently, many improved algorithms are proposed one after another, such as Bootstrap filter (Gordon et al. 1995), Condensation [10] (Isard and Blake 1996), Monte Carlo filters [11] (Kitagawa 1996), Sequential Monte Carlo methods [12] (Doucet et al. 2001), Particle filter [13] (Doucet et al. 2001). These methods boost the validity of particle filtering technique application, and enlarge its application space. However, the framework of particle filtering does not cover the mechanism of data association. When the target number changes or the targets shelter from each other during the process of multi-target tracking, target tracking will fail. Furthermore, the interference among some targets and background random noise will influence the accuracy of tracking.

1.1. Summarize

Based on implementing target tracking by means of particle filtering, a technique framework of tracking target by integrating particle filtering and background modeling is presented. The multi-target tracking is classified into 5 modules as background modeling, multi-target tracking, initializing, re-initializing and particle filtering. The author models each pixel of the image with Gaussian Mixture Model (GMM) in background modeling, to calculate the probability of background pixel in the current image so as to abstract foreground moving objects. When processing the shadow of foreground moving objects, in order to overcome the influence of illumination change of part or integrity in the image, the color model of brightness distortion and chrominance distortion is established to distinguish shaded background, highlight background, original background and foreground objects, and then select the threshold automatically to classify the pixel points in the image. The experiments indicate that this algorithm can distinguish foreground objects from its shadow and increase the efficiency of tracking algorithm.

Based on the background modeling, this article detailed investigates the algorithm flow and technique framework of generating the particle set of each object and particle filtering. Then the algorithm and realizing methods of targets location, target modeling studying, particle initiation, particle resampling, particle updating, weight evaluation and state estimation are presented. In the process of evaluating particle weight, in order to distinguish the different color features of the objects, the original algorithm (evaluating through Bhattacharyya distance) was improved. Only the color distribution of the foreground pixel in particle area after the background modeling is counted, therefore the accuracy and efficiency of target tracking are increased. The experiments prove that this algorithm can realize the effective tracking of multiple moving targets.

1.2. The Main MMT Framework

Fig. 1 is the realization flow frame of multi-target tracking technique based on particle filtering and background modeling, which can be divided into five parts:

1) Background modeling: modeling the background image sequence with GMM algorithm so as to abstract foreground moving objects from background;

2) Initiation: abstracting all targets from initial frame by use of background model, then establishing corresponding target model respectively according each target's appearance;

3) Target tracking: detecting the image boundary area, and estimating either new target appearing or existing target disappearing;

4) Particle filtering: tracking the targets with particle filtering algorithm, studying background model and each target's model. The process includes: particle initiation, particle resampling, particle updating, weight evaluation and state estimation;

5) Re-initiation: when detecting the situation of new target appearing or existing target disappearing, estimating each target's state in the current frame by target orientation and belief theory, and remove or increase changing targets in target model gather.

2. GMM Background Modeling

It is very important to abstract foreground area from background image in image sequence for the following treatment such as target classification, target tracking and behavior comprehension etc. However, background modeling is very difficult because of dynamic changes of background image, such as weather, illumination, shadow, confusion disturbance etc. If vision is fixed and background is

absolutely silent, each pixel of background image can be described with a Gaussian distribution. But background scene is not always absolutely silent. For example, as a result of branch swinging, one pixel in background at a certain time may be a leaf, branch, building, or sky. Besides, the same pixel's chromatic value in image plane may change in different time because it will be effected by beam changing or other factors. Therefore, the actual background cannot be described with only one Gaussian model, but multi-Gaussian model. After acquiring new frame image, the GMM will be updated, each pixel in current image matching with GMM. If the matching is positive, this pixel is background pixel, otherwise it is foreground pixel.

2.1. GMM Algorithm

Suppose the number of Gaussian distribution describing each pixel color is K. The probability function of pixel is given by:

$$P(z_{uv}) = \sum_{j=1}^{K} w_{j,uv}\eta(z_{uv} \mid \mu_{j,uv}, \sum_{j,uv}), \quad (1)$$

where $w_{j,uv}, \mu_{j,uv}$ and $\sum_{j,uv}$ are the weight, mean and covariance of jth Gaussian distribution. The probability function η is defined as:

$$\eta = (X_t, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_t-\mu)^2}{\sigma^2}}, \quad (2)$$

Because we model chromatic image, moreover we consider that R, G, B three channels are independent with each other, so $\mu_{j,uv}, \sum_{j,uv}$ can be defined as follows:

$$\mu_{j,uv} = (\mu^R_{j,uv}, \mu^G_{j,uv}, \mu^B_{j,uv},), \quad (3)$$

$$\sum_{j,uv} = diag((\sigma^R_{j,uv})^2, (\sigma^G_{j,uv})^2, (\sigma^B_{j,uv})^2), \quad (4)$$

Each Gaussian distribution is arranged by priority, we define priority p_j as:

$$p_j = \frac{\omega_{j,uv}}{\sqrt{(R_{j,uv} - \mu^R_{j,uv})^2 + (G_{j,uv} - \mu^G_{j,uv})^2 + (B_{j,uv} - \mu^B_{j,uv})^2}}, \quad (5)$$

where we suppose each color channel be independent with each other due to calculating capacity. The value of K commonly range from 3 to 5. The larger K is, the more complicate scene is described by system. But the calculating capacity will also increase greatly.

2.2. The Realization of GMM Algorithm

Fig. 1 is the realization process of GMM, which includes 3 steps:

1) Model initiation: Taking each pixel of first picture as average value, giving a larger variance and a smaller weighting.

2) Model study: Picking up background scene figure, comparing each pixel with k (k<=K) Gaussian model of this pixel. If $|z - \mu_{j,uv}| < 2.5\sigma$, then regulating the jth Gaussian model parameter and weighting. Otherwise, if k<K, increasing one Gaussian model; if $k = K$, replacing the lowest priority Gaussian distribution with the new one. The new Gaussian distribution taking the z value as average value, giving a larger variance and a smaller weighting. So continually training with background sample picture, the background model of Gaussian mixed distribution is eventually gained.

3) Foreground picture evaluating: Because of the noises and moving targets, some pixels of certain picture in background picture library cannot really represented background, the Gaussian distribution model building with these pixels should be eliminated. Supposing moving targets and noise will not stay at one certain location for a long time, then the weighting and priority of these noises and moving targets' corresponding Gaussian model will be very small. Arranging K Gaussian distributions according to the priority, the first B distributions is taken as background model, the B is defined as follows:

$$B = \min_b (\sum_{j=1}^{b} \omega_{j,uv} > M), \quad (6)$$

where the M is the advanced defined threshold value, which can really reflect the lowest share of background data in total data. If $M = 1$, the background model is single Gaussian distribution; if M>1, the background model is mixed Gaussian distribution.

To every frame image, we compare the each pixel with the separately Gaussian model of the corresponding Gaussian mixed model. If $|z - \mu_{j,uv}| < 2.5\sigma$, this pixel is background pixel; otherwise it belongs to foreground one.

3. Implementation of MMT

The process of multi-target tracking is to estimate the target moving state in real time according to info source data, so as to realize target tracking. The main focus in this chapter is multi-target tracking technique based on particle filtering. This chapter's work is based on background modeling in the above chapter.

Supposing the number of targets is M, the problem of MMT can be considered as the combination of some single target tracking process, namely realizing M single target tracking process with particle filtering. The flow of tracking target $\omega_k (1 \le k \le M)$ with particle filtering, which includes target model studying, particle initiation, particle resampling, particle updating, weight evaluating and state estimating, is given as Fig. 3.

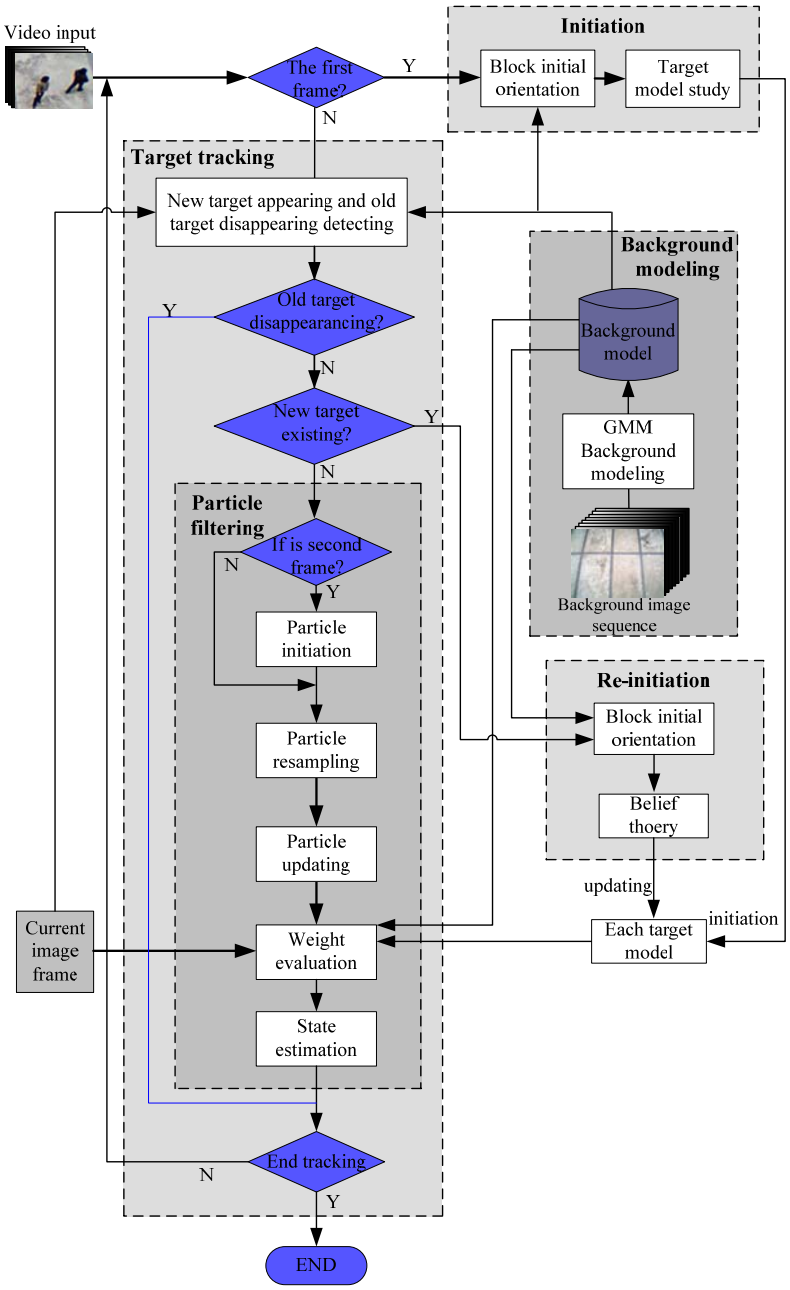


Fig. 1. The realization flow of multi-target tracking based on particle filtering and background modeling.

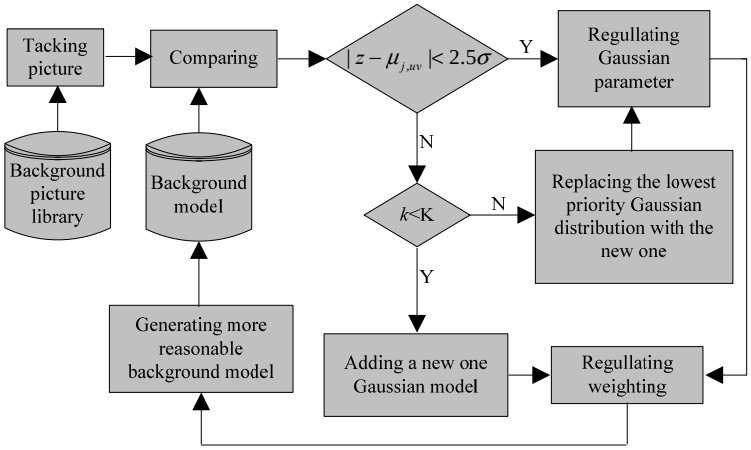


Fig. 2. The realization process of Gaussian mixed background modeling.

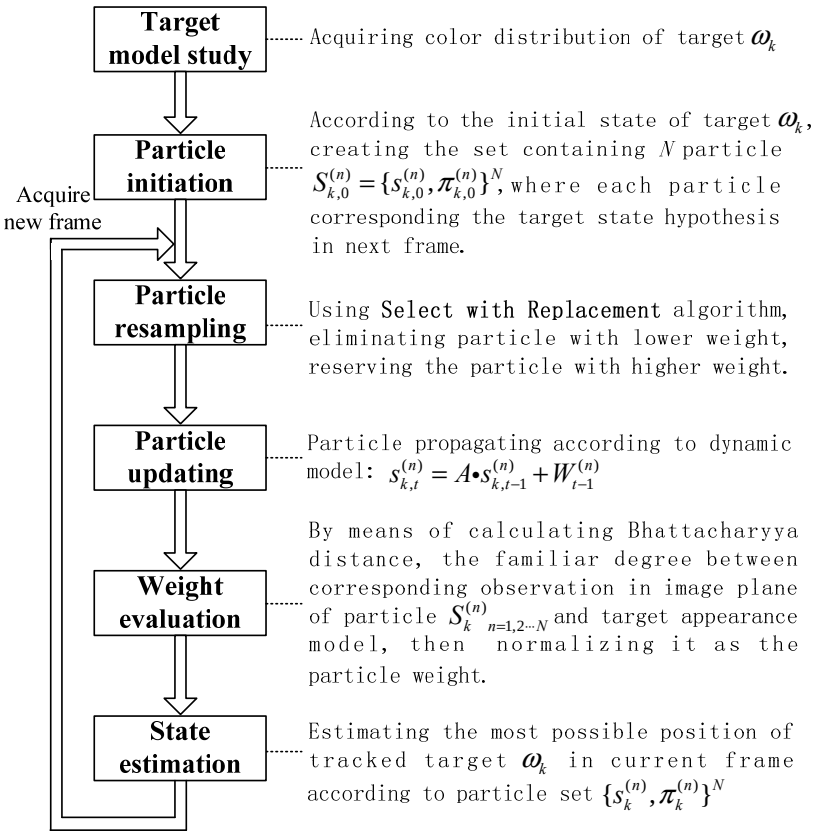


Fig. 3. Flow of tracking target ω_k based on particle filtering.

3.1. System State Description and Model

In this system, we describe the particle state as a fixed-size rectangle, adopt calibrated vidicon furthermore consider all targets satisfying coplanar restriction, namely $z = 0$. So the nth particle state of target ω_k is defined as: $s_{k,t}^{(n)} = \{x, y, \Delta x, \Delta y\}$, where (x, y) are center position of rectangle, $\Delta x, \Delta y$ are respectively denote horizontal and vertical velocity. Our focus are the state of x and y. The dynamic model we adopted is given in (7), where, $w_{t-1}^{(n)}$ is a Gaussian random noise, $w_{t-1}^{(n)} \in N(0, \sigma_s)$, σ_s is particle state covariance.

$$s_{k,t}^{(n)} = A \bullet s_{k,t-1}^{(n)} + w_{t-1}^{(n)}, \tag{7}$$

3.2. Particle Initiation

We firstly utilize the processing result of GMM, locate the original target and obtain the template of target particle ω_k. Then we acquire the initial state of target ω_k, which is the state $s_{k,0}$ of ω_k in it's original appearing time. Then we generate a particle set $s_{k,0}$ in a certain range place near $s_{k,0}$ according to the random mode:

$$S_{k,0} = \left\{ (s_{k,0}^{(n)}, \pi_{k,0}^{(n)}) \mid n = 1 \ldots N \right\}, \tag{8}$$

where N is the particle number, $s_k^{(n)}$ is the supposing state of target ω_k in next moment, $\pi_{k,0}^{(n)} = 1/N$.

3.3. Target Model Study and Weight Evaluation

In this part, the color distribution histogram of particle area is adopted to estimate the particle weight, it's main concept is: comparing each pixel number of RGB chrominance between particle appearance and target model, if the target model and particle appearance are similar, the pixel number difference of corresponding chrominance between particle and target model are also little; otherwise, the pixel number difference are much more larger; then making use of Bhattacharyya distance to detect the difference so as to count the particle weight.

1) Target model study.

After abstracting all targets from initial frame using background model, then corresponding target model according to each target's appearance is separately established using following Bhattacharyya distance algorithm [14]. Ordering m=255, R, G, B three channel color distribution $q = \{q^{(u)}\}_{u=1,2,\cdots,m}$ is built as following target model:

$$q(u) = f\sum_{i=1}^{I} k(\frac{||x_i||}{a})\delta[h(x_i) - u], \quad (9)$$

where I is the number of pixels in the region, δ is the Kronecker delta function. The parameter $a = \sqrt{H_x^2 + H_y^2}$ is used to describe the size of the region, and function is normalization factor, which ensure $\sum_{1}^{m} q(u) = 1$.

$$f = \frac{1}{\sum_{i=1}^{I} k(\frac{||x_i||}{a})}, \quad (10)$$

To increase the reliability of the color distribution when boundary pixels belong to the background or get occluded, smaller weights are assigned to the pixels that are further away from the region center by employing a weighting function, where r is the distance between the pixel and the particle region center.

$$k(r) = \begin{cases} 1 - r^2 & r < 1 \\ 0 & otherwise \end{cases}, \quad (11)$$

We can also acquire the R, G and B three channel color distribution q_R, q_G and q_B by formula (9). In this way, the target color distribution model is built.

2) Particle weighing evaluation.

During the tracing process at different times, to the corresponding observation of every particle state supposing, we always calculate it's appearance probability. Therefore, we need a similar evaluating method to uniformly evaluate the similar degree of each particle appearance with the target model.

In weight evaluation part of particle filtering, we also statistic the R, G, B three channel color distribution of particle with position y according to formula (9): $p_y = \{p_y(u)\}_{u=1,2,\cdots,m}$. Then, the similar degree between $p(u)$ and $q(u)$ is measured by Bhattacharyya coefficient:

$$\rho[p,q] = \int \sqrt{p(u)q(u)}du, \quad (12)$$

Considering discrete densities such as our color histograms $p = \{p(u)\}_{u=1\ldots m}$, $q = \{q(u)\}_{u=1\ldots m}$, the coefficient is defined as follows:

$$\rho[p,q] = \sum_{u=1}^{m} \sqrt{p(u)q(u)}, \quad (13)$$

The larger ρ is, the more similar the distribution is. In this way, we measure three similar degrees between $p(u)$ and $q(u)$ of R, G and B channels ρ_R, ρ_G, ρ_B. Then, integrate ρ_R, ρ_G, ρ_B to acquire the similar degree between particle area and target model:

$$\rho = \sqrt{\rho_R \bullet \rho_G \bullet \rho_B}, \quad (14)$$

After normalization, we calculate the weight of each particle $S^{(n)}$, and ensure $\sum_{n=1}^{N} \pi^{(n)} = 1$.

$$\pi^{(n)} = \frac{\rho^{(n)}}{\sum_{i=1}^{N} \rho^{(n)}}, \quad (15)$$

3) Improved algorithm based on background modeling.

We analyze particle weight evaluation method based on Bhattacharyya distance in the above section. But this algorithm statistic all pixel color distribution in particle area, which does not divide foreground pixel and background pixel. In this way, if color distribution of particle area background pixel is very abundant, or is similar to color distribution of particle area foreground pixel, the calculated particle weight is also very large. Thereby the accuracy of particle weight evaluation is reduced so as to influence the well-balanced implementation of other step such as particle resampling, particle updating, and state estimation etc. Hereon, we present the following improved algorithm.

Firstly, on the base of background modeling in above chapter, the binary image processed by background modeling is defined as follow:

$$\xi(x_i) = \begin{cases} 1 & x_i \text{ is foreground pixel} \\ 0 & x_i \text{ is background pixel} \end{cases}, \quad (16)$$

After that, we just only statistic foreground pixel color distribution of particle area, without consideration of background pixel. Namely revising formula (9) as follows:

$$q(u) = f\sum_{i=1}^{I} k(\frac{||x_i||}{a}) \bullet \delta[h(x_i) - u] \bullet \xi(x_i), \quad (17)$$

In this way the optimized algorithm not only accurately and effectively calculate particle weight, but also condense the program running time of weight evaluation part so as to increase the efficiency of whole particle filtering process. The strongpoint of improved algorithm will be analyzed through experiment result in details in chapter 4.

3.4. Particle Resampling

During the realization process of particle filtering, the degradation phenomena will happen in some time, which means many particle weight will become very small through several step of iteration. Because the weight variance will increase with tracing time, the degradation phenomena is invertible. It means that a mass of calculation will waste to the particles with tiny weight, but these particles' contribution to the state evaluating is almost zero. While the particle resampling can figure it out preferably.

At present, the methods of decreasing particle weight variance are approximately several categories as follow [15]: 1) Sampling Importance Resampling SIR; 2) Residual Resampling; 3) Markov Monte Carlo method; 4) Rao-Blackwellised method.

3.5. Particle Updating

According to the particle dynamic model (formula 7), the particle state of new frame is forecasted, which is multiplying and delivering the resampled repeatedly particles with higher weight.

3.6. State Estimation

After abstaining all particles' weight, the next step is to forecast the most probable appearing position of traced target. The detailed method we employed is that calculating the weighted average of all the particles according to the each particle's weight, as target state, that is estimate the target state using following formula:

$$E[s_t] = \sum_{n=1}^{N} \pi_t^{(n)} s_t^{(n)} , \qquad (18)$$

4. Experiment Result Analysis

Thus we discuss the GMM background modeling, each procedure of MMT based on particle filtering and background modeling[16], and disposal methods of new target existing and old target disappearing. We applied the algorithm to detecting and tracking certain passerby in outdoor.

4.1. Result of GMM

Fig. 4 presents the GMM experiment results by choosing different threshold and different Gaussian distribution number *K*. Fig. 4(a) is the result after equalizing *N* background figure; Fig. 4(b) is foreground figure.

Fig. 4(c) and Fig. 4(d) is GMM result with the same threshold but using different *K* value (*K* =2 and 7), which verify that the larger *K is*, the more complicated scene the system can represent, and the better the GMM effect is, the calculated amount will also sharply increase.

Fig 4(e) and 4(f) is GMM result with the same *K* value but using different threshold, which verify that the smaller threshold value, the removed pixels are more, and the GMM effect is better. While the threshold cannot be set to too small, otherwise some certain foreground pixels will be removed after GMM, then effect the feature extraction of target detection.

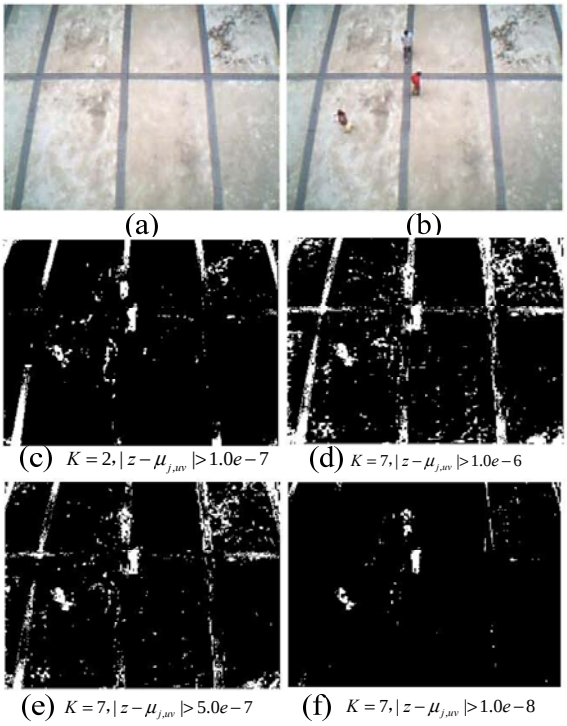


Fig. 4. The GMM results by choosing different threshold and different *K*.

4.2. Result of Weight Evaluation

In section 3 the weight evaluation algorithm with Bhattacharyya distance is analyzed, moreover, an improved algorithm is presented. Comparison between two algorithms is analyzed with the following experiment data.

1) **Experiment result of two weighing evaluation algorithm and analysis.**

Fig. 5 is the experiment result directly using Bhattacharyya distance to evaluate the weight of target particle A. Fig. 5(a) is the weight calculating result of particle A. Fig. 5(b), Fig. 5(c) and Fig. 5(d) are separately the R, G, B three channel color distribution figure of particle and target model, where red represent color distribution *p* of particle, and blue represent color distribution *q* of target mode.

Fig. 6 is the result using improved algorithm to evaluate the weight of target particle A. Where sub Fig. 6(a) present the appearance of target model and particle, as well as weight calculating result according to formula (13) and (14), including $\rho_R[p,q]$, $\rho_G[p,q]$, $\rho_B[p,q]$ and ρ , where $\rho=\sqrt{\rho_R \cdot \rho_G \cdot \rho_B}$. For the comparing convenience, the figures present the result after weight normalizing. Sub Fig. 6(b), Fig. 6(c) and Fig. 6(d) are separately the R, G, B three channel color distribution figure of particle *q* (red) and target model *p* (blue).The two group of experiment results show that the R, G, B three channel color distribution figure of particle *q* (red) is more closely to the target model *p* (blue) by means of the improved algorithm

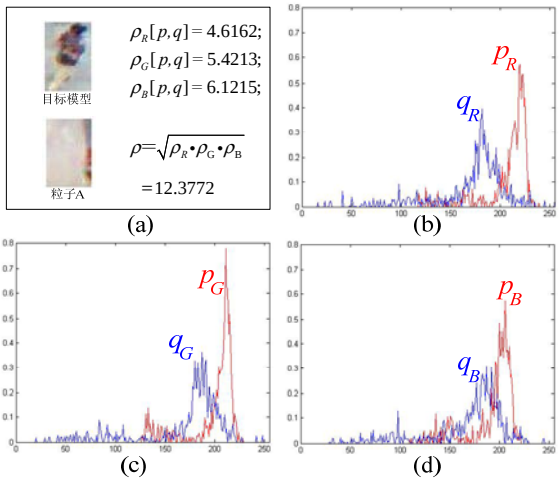


Fig. 5. The result using Bhattacharyya distance to evaluate the weight of particle A.

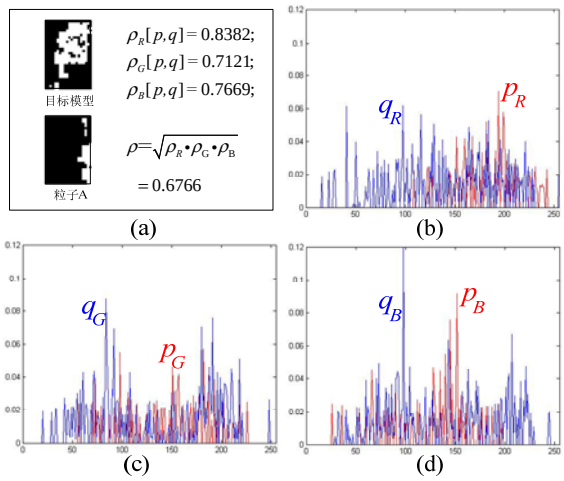


Fig. 6. The result using improved algorithm to evaluate the weight of particle A.

2) Experiment of improved algorithm

In this section, we present the 20 particles area image of the target and the comparing result between two algorithms. Where, the sign in the top left corner of each particle image is the particle serial number, $w1$ is the original algorithm result, and $w3$ is the improved algorithm result.

Through Fig. 7, the weight evaluation between particle ω_6 and ω_{18} can be clearly compared. ω_6 basically does not cover the foreground target, while ω_{18} relatively integrated and covering the foreground target. The weight of two particles should be obviously different.

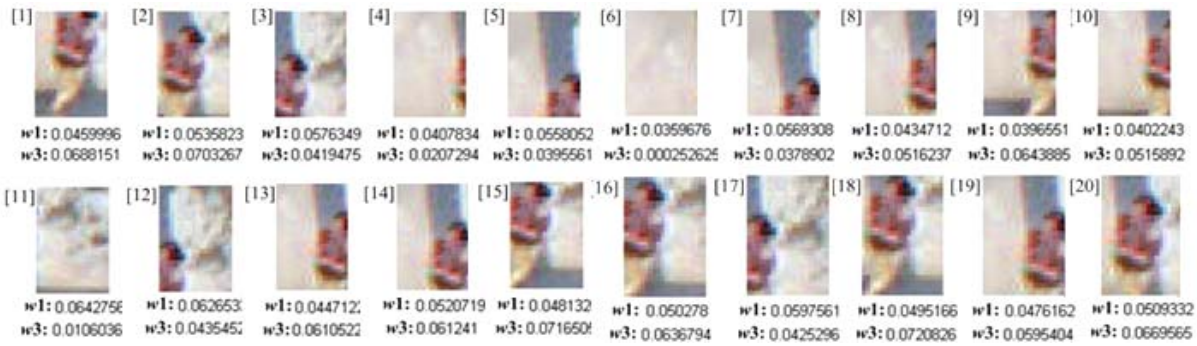


Fig. 7. 20 particles area image of the target and the comparing result between two algorithms (where w1 is the original algorithm result, and w3 is the improved algorithm result).

Table 1 shows the comparison result of weight evaluation between ω_6 and ω_{18} with two algorithms. The last column of table 1 is the ratio of π_{18} to π_6.

Table 1. Comparison result of weight evaluation.

Algorithm	Weight π_6	Weight π_{18}	π_{18}/π_6
Bhattacharyya distance algorithm	0.0359	0.0495	1.37
Improved algorithm	0.000252	0.0720	285.33

It can be seen from table 1 that there is a severe error if Bhattacharyya distance is used directly to evaluate the particle weight and result in target loss in following frame tracking; while the improved algorithm experiment result is relatively ideal, and exactly evaluate these two particles: particle ω_6 is an obvious error area, its weight is very small; while particle ω_6 almost cover the whole foreground target and has a large weight, which is 285 larger than ω_6's so as to effectively distinguish these two particles.

Synthesizing the above comparison and analysis, we can conclude that the error of the improved algorithm is smaller; the weight evaluating is more precise. Moreover the program running time of weight evaluating is shortened because only the foreground pixels is statistic but not the background pixels, so as to improve the efficiency of whole particle filtering.

4.3. Result of Multi-target Tracking

Thus we discuss every step of target tracing based on particle filtering, and compare the effect of each implementation method in some steps. Then the realization and experiment result of target tracing is given as below.

We use the above algorithm to trace multi-targets in outdoors scene, sequence image resolution is 320×240. Fig. 8 is the result of tracking three targets (particle number N=50). In Fig.8, blue, green and white lines respectively denote three target movement locus. In frame 20, the first target appears; in frame 120, the second target appears; in frame 187, the third target appears; in frame 300, the first target disappears; in frame 311, the last two targets disappear.

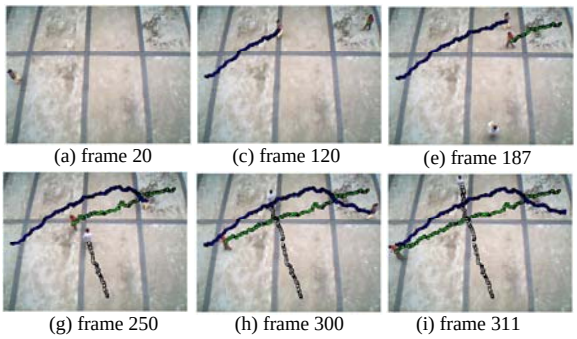


Fig. 8. Result from tracking three targets (N=50).

Fig. 9 is the result from tracking certain target when choosing different particle number N, where black locus is actual calibrated moving focus, while white one is the particle filtering tracking result.

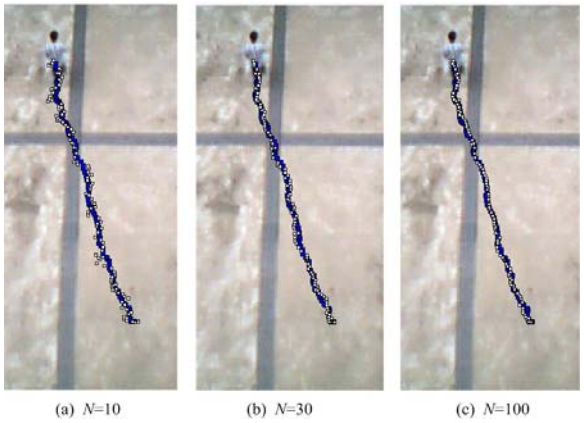
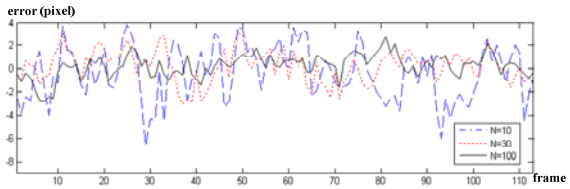


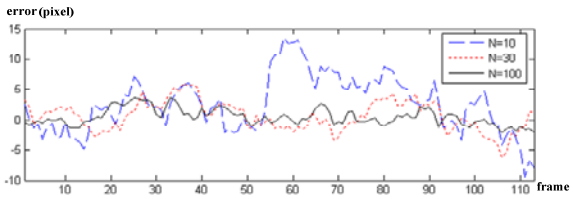
Fig. 9. Tracking result comparison choosing different particle number N (blue locus is actual calibrated moving focus, white one is the particle filtering tracking result).

When N=10, the approximate tracking focus can be seen but with relatively big error; when N=30, the tracking error is obviously smaller than the situation of N=10; when N=100, the tracking focus is best closer for actual target moving focus, and with the least tracking error, but this situation will consume the most time for program running.

Fig. 10 is horizontal and vertical direction error analysis when choosing different particle number N to tracing target 3. It also validates the relationship between particle number and tracking error. If the particle number is less, tracing error will become bigger but program running time consuming will be less. Otherwise, along with the enlargement of particle number, tracing precision will greatly improve but consuming more program running time.



(a) Horizontal direction



(b) Vertical direction

Fig. 10. Error analysis when choosing different particle number N.

The above result illustrate that we should regulate particle number N during targets tracing, according to different circumstance, so as to improve the efficiency of MMT.

Fig. 11 is the tracing result in frame 230 (N=50). Where the red, green and yellow are separately particle set tracing target 1, 2 and 3. The blue is state estimating result.

Fig. 11. The tracing result in frame 230 (N=50).

5. Conclusion

Based on particle filtering algorithm, a technique framework of tracking target by integrating particle filtering and background modeling is presented. Compared with other tracking algorithm, this algorithm has the following innovation: 1) The multi-target tracking (MMT) is classified into 5 modules as background modeling, multi-target tracking, initializing, re-initializing and particle filtering; 2) Based on the background modeling, the algorithm flow and technique framework of generating the particle set of each object and particle filtering are presented; 3) In the process of evaluating particle weight, the original algorithm (evaluating through Bhattacharyya distance) was improved. Only the color distribution of the foreground pixel in particle area after the background modeling is counted, therefore the accuracy and efficiency of target tracking are increased; 4) By means of the results of background modeling, the connection between time and space of the moving targets, and belief theory, the problem of new target appearing and old targets disappearing can be solved. The experiments prove that this algorithm can realize the effective tracking of multiple moving persons.

References

[1]. K. P. Karmann, A. von Brandt, Moving object recognition using an adaptive background memory, *Time-varying Image Processing and Moving Object Recognition*, *Elsevier Science Publishers B.V.*, Vol. 2, 1990, pp. 234-239.

[2]. M. Kilger, A shadow handler in a video-based real-time traffic monitoring system, in *Proceedings of IEEE Workshop on Applications of Computer Vision, Palm Springs,* CA, 1992, pp. 1060-1066.

[3]. Thanarat Horprasert, David Harwood, Larry S. Davis, A statistical approach for real-time robust background subtraction and shadow detection, *IEEE Frame Rate*, Vol. 7, 1999, pp. 808-830.

[4]. C. Stauffer, W. Grimson, Adaptive background mixture models for real time tracking, in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, Vol. 2, 1999, pp. 246-252.

[5]. Fang Shuai, Tong Shi-Zhong, A method of target detection and tracking in video surveillance, in *Proceedings of the 3rd International Conference on Image and Graphics ICIG*, 1999, pp. 232-239.

[6]. Y. Satoh, T. Okatani, K. Deguchi, A color-based tracking by Kalman particle filter, in *Proceedings of the 17th International Conference on Pattern Recognition ICPR'04*, Vol. 3, 2004, pp. 502-505.

[7]. M. Isard, and A. Blake, Condensation – conditional density propagation for visual tracking, *International Journal of Computer Vision*, Vol. 29, No. 1, 1998, pp. 5-28.

[8]. V. Pavlovic', J. Rehg, T. J. Cham, K. Murphy, A dynamic Bayesian network approach to figure tracking using learned dynamic models, in *Proceedings of* the *IEEE International Conference on Computer Vision, Corfu, Greece,* 1999, pp. 94-101.

[9]. M. Sanjeev, Simon Maskell, Neil Gordon, Tim Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 26-76.

[10]. Isard, M., Blake, A. A Mixed-State condensation Tracker with Automatic Model-switching. in *Proc. of the IEEE Int'l Conf. on Computer Vision*, 1998, pp.107-112.

[11]. Jaco Vermaak, Patrick Perez, Monte Carlo filtering for multi-target tracking and data association, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 1, 2005, pp. 309-330.

[12]. Augusting Ooi, Ba-Ngu Vo, Arnaud Doucet, Particle filtering for multi-target racking using jump Markov systems, in *Proceedings of the IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 14-17 December 2004, pp. 131-136.

[13]. Hedvig Sidenbladh, Multi-target particle filtering for the probability hypothesis density, in *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Australia, 2003, pp. 800-806.

[14]. P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, pp. 564-575.

[15]. Ioannis Reckleitis, A particle filter tutorial for mobile robot localization, in *Proceedings of* the *International Conference on Robotics and Automation*, Vol. 42, 2003, pp. 1-36.

[16]. Liao Xuechao, Research and Implementation of Multi-target Tracking Technique Based on Particle Filtering and Background Modeling, *Wuhan University of Science and Technology*, 2006.