TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 10, October 2014, pp. 7381 ~ 7388
DOI: 10.11591/telkomnika.v12i8.5342

Anti-Occlusion Algorithm for Object Tracking Based on Multi-Feature Fusion

Jie Cao, Liang Xuan*
School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, P.R.China
*Corresponding author, email: xy2771753@163.com

Abstract

Complex background, especially when the object is similar to the background in color or the target gets blocked, can easily lead to tracking failure. Therefore, a fusion algorithm based on features confidence and similarity was proposed, it can adaptively adjust fusion strategy when occlusion occurs. And this confidence is used among occlusion detection, to overcome the problem of inaccurate occlusion determination when blocked by analogue. The experimental results show that the proposed algorithm is more robust in the case of the cover, but also has good performance under other complex scenes.

Keywords: *computer application, object tracking, anti-occlusion, particle filter, multi-feature, template update*

1. Introduction

Video object tracking is an important research direction of computer vision that is widely used in vedio surveillance, human-computer interaction, robot visual navigation, medical diagnostics and other areas. Target is often blocked by non-target obstacle and other target in complex background. How to deal with occlusion is a problem that should be solved quickly in current visual tracking areas.

References [1-2] have proposed a kalman filter algorithm to predict target position, but it is likely to fail when target moves erratically or is blocked for a long time. Reference [3] has used the difference of pixel gray predicted values and measurements to determine if an occlusion, but it did not apply to the situation that the target trajectory changed greatly. Reference [4-5] have proposed a tracking algorithm based on probabilistic appearance model, which timely updated the target color model to keep the target color spatial distribution information, but it can not track target when target is similar to shelter. Reference [6] represented the target with multiple different windows, then used the position relation and similarity of the windows to estimate the true position of the target, it can solve partial occlusion, but be interfered by background noise. References [7-9] solved the occlusion by blocking the target, but lacked of stability and were low efficiency in complex background. Reference [10] has proposed an occlusion detection method based on adaptive fusion coefficient, but there is a big error when the weight distribution is relatively uniform.

As the fixed fusion methods will reduce the tracking performance when the object is occlusion, so we need adaptive fusion strategy to reduce its influence on the tracking results. In this paper, using the confidence dynamically adjusts the fusion weights of each feature in the framework of particle filtering, and it can be used to block detection. According to the change of total similarity, the fusion method and template update strategy are dynamically adjusted, which can reduce the effect of noise. The experiments illustrate the effectiveness and superiority of the algorithm.

2 Algorithm Principles

2.1. Particle Filter

Particle filter is an effective technical means and tools in solving non-Gaussian and non-linear system state tracking problem. It calculates recursively mainly through the prediction and update:

Received December 11, 2013; Revised July 18, 2014; Accepted August 2, 2014

1) The prediction process

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1})\, p(x_{t-1} \mid z_{1:t-1})\, dx_{t-1} \tag{1}$$

2) The update process

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1})}{\int p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1})\, dx_t} \tag{2}$$

For the system of non-linear and non-Gaussian, formula (1), (2) integral cannot be calculated analytically. Particle filter uses a set of weighted random sampling point to approximate to the target state posterior probability distribution, when the sampling points approaches infinity, it can obtain the optimal Bayesian estimation of target states. The posterior probability distribution can be approximately expressed as:

$$p(x_t \mid z_{1:t}) = \sum_{j=1}^{N} w_t^j \delta(x_t - x_t^j) \tag{3}$$

Where N is the number of particles; w_t^j is the the j particle weight at time t, which satisfies the normalization condition $\sum_{j=1}^{N} w_t^j = 1$; $\delta(\cdot)$ is dilta function. After introducing the importance density function $q(\cdot)$ in the particle sampling, the weights recursive equations:

$$w_t^j \propto w_{t-1}^j \frac{p(z_t \mid x_t^j)\, p(x_t^j \mid x_{t-1}^j)}{q(x_t^j \mid x_{t-1}^j, z_{1:t})} \tag{4}$$

Choosing prior distributions $p(x_t \mid x_{t-1})$ as the importance sampling function, the weight updating process (4) can be simplified as:

$$w_t^j \propto w_{t-1}^j p(z_t \mid x_t^j) \tag{5}$$

The target state estimate value is the average weighted of particle set in time t:

$$\hat{x}_t = \sum_{j}^{N} w_t^j x_t^j \Big/ \sum_{j}^{N} w_t^j \tag{6}$$

Because the motion characteristics of any targets is difficult to obtain, this paper choose one order linear system which is the most commonly used as the particle filte state transfer modelr:

$$x_t = Ax_{t-1} + W \tag{7}$$

Where A is state transition matrix, which is the unit matrix in here, W is Gauss noise with 0 average.

2.2. Feature Model

The basic idea to achieve multi feature tracking in the framework of particle filter is that, Firstly, describe the multi feature model of target area $Q = \{q_i\}_{i=1,2,\cdots,n}$, where i represents different feature space, $q_i = \left\{ q_{u_i}\left(\hat{y}\right) \right\}_{u_i=1,2,\cdots,m_i}$ are each feature sub-models described with the weighted kernel function histogram. Usually expressed as:

$$q_{u_i}\left(\hat{y}\right) = C_h \sum_{l=1}^{B} K\left(\frac{\left\|\hat{y} - x_l\right\|}{a}\right)\delta\left[b_i\left(x_l\right) - u_i\right] \tag{8}$$

Where $\hat{y}$ is the center position of target area; B and a represent the pixel number and scale of the target area; δ is delta function, $b_i\left(x_l\right)$ denotes corresponding histogram index values of the pixel color in image x_l, u_i is the sub interval of each feature space; $K\left(r\right)=1-r^2$ is a kernel function, which sets the smaller weights to the pixels away from the target template center, because these pixels are susceptible to the other goal conflict or background pixels.

Then set the particle concentration of each particle area as the target candidate, establish the corresponding candidate model $P^j = \left\{ p_i^j \right\}_{i=1,2,\cdots,n}^{j=1,2,\cdots,N}$, where j are particles, $p_i^j = \left\{ p_{u_i}\left(y_j\right) \right\}_{u_i=1,2,\cdots,m_i}$ is the i feature sub model of candidate model, y_j is the central position of a particle, and use Bhattacharyya coefficient to measure the similarity of particle area and each sub model:

$$\rho_i^j = \sum_{u_i=1}^{m_i} \sqrt{p_{u_i}\left(y_j\right) q_{u_i}\left(\hat{y}\right)} \tag{9}$$

On this basis, calculate particle observation probability of each feature at time t:

$$p\left(z_i^j \mid x_i^j\right) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right) \tag{10}$$

Where σ is the variance of the Gauss distribution; $d_{i,j} = \sqrt{1 - \rho_i^j}$.

3. Fusion and Occlusion

In the actual tracking process, because of the influence of noise, sensor instability or other interference factors, using a fixed combination method will reduce the tracking performance. So we need a adaptive fusion strategy to reduce its influence on the tracking results. This paper analyzes the spatial distribution and weights distribution of particles to measure each feature confidence in the tracking process, as a fusion factor of the observation probability and Bhattacharyya coefficient of block determination.

3.1. Feature Fusion

Firstly, we use variances to measure the spatial distribution of particles, and select a certain proportion of each feature particles as the reference particles after resampling, in order to prevent individual particles from migrating farther, caused by excessive variance. The smaller the variance of the position distribution is , more concentrated the feature particles are, the smaller the uncertainty is.

Define 1: The variance of the position distribution can be defined as:

$$\sigma_i = \frac{1}{m}\sum_{j=1}^{m}\left(x_i^j - \overline{x}_i\right)\left(x_i^j - \overline{x}_i\right)^{\mathrm{T}} \tag{11}$$

Where $\overline{x}_i$ is the mean of the sample position, m is the number of particles, x_i^j is the j-th particle of the i-th feature.

Then we use the observed entropy to measure the weight distribution, greater entropy is, more uniform the weight distributio is, weaker identification ability is.

Define 2: The weight distribution can be defined as:

$$H\left(p_i\right) = -\sum_{j=1}^{N}\left(p\left(z_i^j \mid x_i^j\right)\log_2 p\left(z_i^j \mid x_i^j\right)\right) \tag{12}$$

The feature weight that spatial distribution is more concentrated and distinguish good on the target should be larger, so we define each feature tracking performance evaluation $h_i = 1/\Delta_i$.

Define 3: The i-th feature fusion weights can be defined as:

$$\lambda_i = h_i \Big/ \sum_{i=1}^{n} h_i \tag{13}$$

Define 4: The new fusion strategy can be defined as:

$$p_t^j = \rho_{t-1}\prod_{i=1}^{n} p_{i,t}^j + \left(1-\rho_{t-1}\right)\sum_{i=1}^{n}\lambda_i p_{i,t}^j \tag{14}$$

Where ρ_{t-1} is the total similarity of target template for a moment.

In this paper, the fusion strategy essentially unifies multiplicative fusion, weighted fusion into an adaptive framework, and dynamically adjusts each feature proportion in the observation probabilities. The method can adapt to environmental changes, to achieve robust tracking better.

3.2. Occlusion Detection and Model Update

Target or scene is likely to change in the tracking process. Using a single fixed target model can not be a long and stable tracking, and the sensitivities of different features to scene change are different, and fixed weights updating method is difficult to meet the requirements. So this paper combines with feature confidence and Bhattacharyya coefficient to calculate the total similarity of target template, and dynamically adjusts update strategy according to its change.

As target is blocked by the analogue, similar characteristics still maintain a high similarity, so are not easy to estimate the occlusion. But at this moment the feature particles are more dispersed, weights distribution is relatively uniform, confidence is lower. So this algorithm uses it to weaken the interference to the occlusion judgment caused by the similar feature.

Define 5: Combined with formula (9) and (13) the total similarity of target template can be defined as:

$$\rho = \sum_{i=1}^{n}\lambda_i\rho_i \tag{15}$$

When the total similarity decreases greatly, the occlusion occurs, at this moment if still updating model, it will incorporate more background noise, causing the error for tracking after occlusion recovery. So at this point we should use the target model a moment ago.

Define 6: The new template update strategy can be defined as:

$$q_t = \begin{cases} q_{t-1} & \rho_t < T \\ \beta q_{init} + (1-\beta)\,p_c & \rho_t \geq T \end{cases} \tag{16}$$

Where $p_c = \beta q_{t-1} + (1-\beta)\,p_{cur}$, β is the similarity of initial template and the current template, q_{init} is the initial template, p_{cur} is the current template, T is the threshold used to determine occlusion, generally 0.6. when occlusion is not occurred, template update combines with the information of initial template、the previous template and current template, which can adapt to the change of other complex environment better.

The algorithm uses features confidence to weaken the interference to the occlusion judgment caused by similar characteristics, and update target template based on the change of the total similarity in order to ensure the accuracy of tracking.

3.3. Algorithm Implementation

We select color and edge features to track target, where the color feature uses 16×16×16 RGB space, the edge feature is described by weighted gradient direction histogram. Specific steps are as follows:

(1) Initialization. Manually select target tracking, extract sub-models of each feature histogram based on formula (8), make $w_{0,i} = 1/N$, $\rho_0 = 0.8$;

(2) Forecast. Calculate the current frame state based on formula (7) and the previous frame state;

(3) Calculation of confidence. Calculate features confidence λ_i by formula (13);

(4) Feature fusion. Calculate particles weights on the basis of formula (14), then estimate target state $\hat{x}_t$ by formula (6);

(5) Template update. Update target template on the basis of formula (16);

(6) Resampling, return step (2).

4. Experimental Results and Analysis

In order to test the performance of this algorithm, we select relevant video sequences for testing. All algorithms are implemented by Matlab on the PC processor Intel 2.3G 2.3G, 1G memory, target position and initial mode are set manually, video sequences are all derived from the standard video library [11]. The experimental data and algorithm parameter are given in Table 1 and Table 2, where HOE is the edge direction histogram.

Table 1. Video Sequences Property

Video	Image size	Target size	Sequence length
1	384×288	32×72	100
2	384×288	32×48	140

Table 2. Algorithm Parameter

The number of particles N	100	Features	Color、HOE
Feature digitalizing	16	Resampling	60%N
Occlusion threshold T	0.6		

In Experiment 1, video sequence is selected from the channel in front of a shopping mall, the target is affected by the illumination and occlusion. From Figure 1, the reference [12] uses fixed fusion, although color characteristics can not accurately identify the target in the intense illumination environment from 10-th frame to 24-th frame, it can still effectively track the target fused the edge feature based on gradient direction; After occlusion (26-th frame), due to still maintain the original fusion methods, enlarge the background noise, caused that most particles move to the similar background target, the track fails. For the proposed method, two feature confidence will be adaptively adjusted in the intense illumination environment, the fusion result is more accurate, the confidence curve is shown in Figure 2; it adaptively reduce multiplicative fusion weight according to the change after occlusion occurs(26-th frame) of total similarity reduce weight fusion, not amplify noises a lot, the fusion result can effectively separate target and pedestrians; After occlusion is over(64-th frame), fusion strategy can adaptively reduce additive fusion weight, and improve the tracking reliability. To further illustrate the effectiveness of the proposed algorithm, we make a quantitative comparison of two algorithms errors. From Figure 3, we can find the proposed method is more effective in the illumination and occlsion environment.

Figure 1. Some Results on Experiment 1 by using Literature [12] Method (the first row) and Proposed Method (the second row) (Frames: 10, 26, 45, 64, 86)

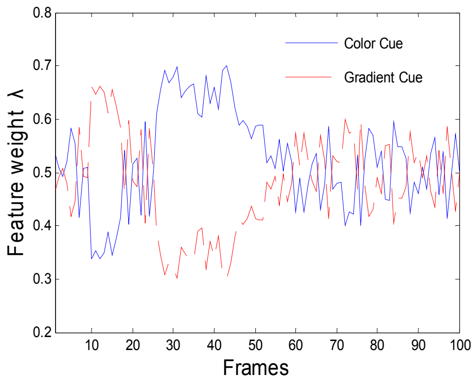

Figure 2. The Update Process of Feature Weights

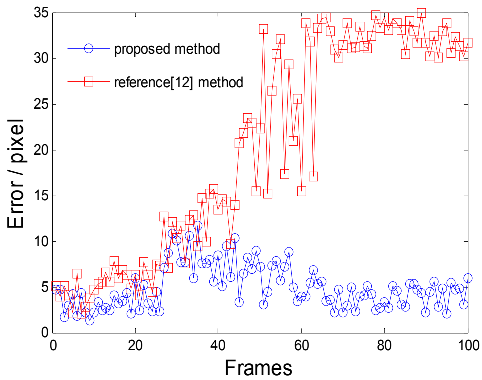

Figure 3. The Error Curve on Experiment 1

Experiment 2 mainly verifies algorithm tracking performance in the shadows and analogue occlusion. From Figure 4, when the target moves into the shadow in 85-th frame, the trackings of sum rule and product rule drift, however, the proposed algorithm tracks accurately. This is because particles diverge, the accuracy of sum rule become lower, and the product rule enlarge the background information. The proposed algorithm combines the advantages of both, and reduces the color weight, so it reduces the effect of shadows on the results, and suppress background noise better; When target is blocked by a similar object, the posterior probability of sum rule can not converge, tracking error is improved distinctly, since the product rule integrates more background noise, losing the target finally. However, the proposed method uses the total similarity to regulate the propotion of sum rule and product rule adaptively, and uses the confidence to allocate the features weights in sum rule reasonably, and ultimately makes the result more accurate and effective. The similarity curve is shown in Figure 5, when the target is blocked by similar object, as the gray information of target and pedestrian is very similar, color feature can not determine ocllusion, however the total similarity can still detect the occlusion accurately, and stop updating the template. From Figure 6, we can find the proposed algorithm is more accurate and stable.

Figure 4. Some Results on Experiment 2 by using Sum Rule (the first row), Product Rule (the second row) and Proposed Method (the third row) (Frames: 68, 85, 98, 104, 117)

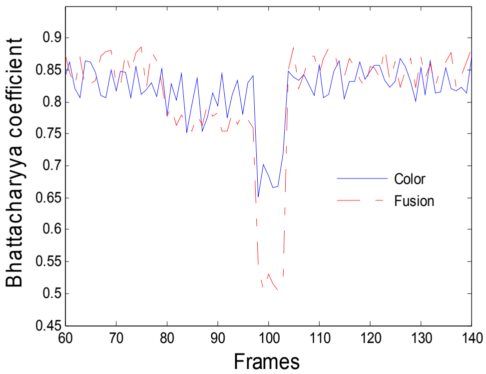

Figure 5. The Bhattacharyya Coefficient Curve

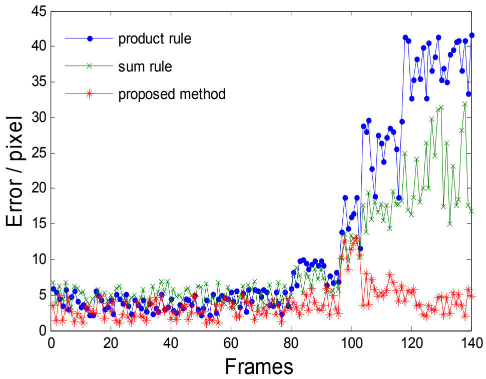

Figure 6. The Error Curve on Experiment 2

In the aspect of real-time, out algorithm time-consuming is mainly in the feature extraction and fusion, but also related to the target size and particles number. Table 3 shows the different tracking speed (fps). The proposed algorithm complexity is similar to sum rule and product rule.

Table 3. The Computational Cost of Four Algorithms

Video	reference[12]	Sum rule	Product rule	Proposed method
1	9.9	10.1	10.2	9.7
2	10.9	11.2	11.4	10.6

5. Conclusion

Video target tracking is widely used in robot navigation, medical diagnosis and video surveillance etc. But because of complex background, target is easy to be blocked analogues, using the fixed fusion method may reduce the tracking performance, therefore, this paper dynamically adjusts the contributions of different features and different fusion methods to the result by using feature confidence and similarity, and applies the confidence into occlusion detection, ensuring the determination result is accurate and effective; When occlusion occurs, we change the template update strategy in time, making template information more accurate. Experiments showes that the algorithm not only has good tracking performance in analogue occlusion, but also can track target effectively in the illumination, shadow and other complex environments. This paper has some limitations on the long time occlusion, the next step is to research the long time full occlusion.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61263031). The Finance Department Foundation of Gansu Province, China (0914ZTB148).

References

[1] Gan MG, Chen J, WANG YN, et al. A target tracking algorithm based on mean shift and normalized moment of inertia feature. *Acta Automatica Sinica*. 2010; 36(9):1332-1336.
[2] Karavasilis V, Nikou C, Likas C. *Visual tracking by adaptive Kalman filtering and mean shift.* Lecture Notes in Computer Science, Artificial Intelligence: Theories, Models and Applications. 2010; 6040: 153-162.
[3] Senior A, Hampapur A, Tian Yingli, et al. Appearance models for occlusion handling*. Image and Vision Computing*. 2006; 24(11): 1233-1243.
[4] Gross R, Mattews I, Baker S. Active appearance models with occlusion. *Image and Vision Computing*. 2006; 24(6): 593-604.
[5] Khan MS, Shah M. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2009; 31(3): 505-518.
[6] An GC, Zhang FJ, Wang HA, et al. Multi-window target tracking. *Journal of Computer Research and Development*. 2011; 48(11): 2023-2030.
[7] Phadke G. *Robust multiple target tracking under occlusion using fragmented mean shift and Kalman filter*. International Conference on Communications and Signal Processing. 2011; 2: 517-521.
[8] Yan J, Wu MY, Chen SZ, et al. Anti-occlusion tracking algorithm based on mean shift and fragments. *Optics and Precision Engineering*. 2010; 18(6): 1413-1419.
[9] Qi MB, Zhang L, Jiang JG, et al. Target template update method in fragment tracking. *Journal of Image and Graphics*. 2011; 16(6): 976-982.
[10] Li YZ, Lu ZY, Li J. Robust video object tracking algorithm based on multi-feature fusion. *Journal of Xidian University*. 2012 ; 39(4): 1-6.
[11] Fisher R. Caviar Test Case Scenarios [DB/OL]. [2011-12-05]. groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.
[12] Birchfield S. *Elliptical head tracking using intensity gra-dients and color histograms.* Proc of IEEE computer society conference on computer vision and pattern recognition. Santa. IEEE.1998: 232-237.