Particle filter based visual tracking with multi-cue adaptive fusion

Anping Li (李安平), Zhongliang Jing (敬忠良), and Shiqiang Hu (胡士强)

Institute of Aerospace Information and Control, School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200030

Received November 26, 2004

To improve the robustness of visual tracking in complex environments such as: cluttered backgrounds, partial occlusions, similar distraction and pose variations, a novel tracking method based on adaptive fusion and particle filter is proposed in this paper. In this method, the image color and shape cues are adaptively fused to represent the target observation; fuzzy logic is applied to dynamically adjust each cue weight according to its associated reliability in the past frame; particle filter is adopted to deal with non-linear and non-Gaussian problems in visual tracking. The method is demonstrated to be robust to illumination changes, pose variations, partial occlusions, cluttered backgrounds and camera motion for a test image sequence.

OCIS codes: 150.0150, 100.0100, 100.2960.

Visual tracking has become a popular topic in the field of computer vision. Its potential applications include smart surveillance, virtual reality, perceptual interface, video conferencing, etc. Although visual tracking has been intensively studied in the literature, developing a robust tracking algorithm in complex environments is still an open problem.

Visual tracking can be considered to match coherent relations of image features between frames. In the last decades, various tracking algorithms have been proposed[1−4]. However, most of them are based on a single image cue. It is clear that no single image cue can be robust enough to successfully deal with various conditions occurring in the real-word scenarios. To overcome the weak robustness of single-cue tracking, many algorithms have been proposed based on multi-cue fusion[5,6]. Multiple cues fusion not only can provide more reliable observation when estimating a state, but different cues may be complementary in that one may succeed when another fails. The key challenge for this kind of algorithm is how to optimally fuse multiple cues. In most algorithms[7,8], the fusion scheme is non-adaptive, in which the reliability of each cue is assumed to be unchanged during the tracking. Such assumption is often invalid due to the dynamically changing environments. To overcome this problem, a novel tracking method based on adaptive fusion and particle filter is proposed.

The tracked target is a human head in this paper. In general, the human head can be approximated by an ellipse. So we use an ellipse to represent the target. The target state is denoted by $X = \{x, y, \dot{x}, \dot{y}, H_x, H_y, \dot{H}_x, \dot{H}_y\}$, where x, y denote the centeriod of the ellipse, $\dot{x}$, $\dot{y}$ the velocities of the centeriod, h_x, h_y the length of two half axes, $\dot{h}_x$,$\dot{h}_y$ the velocities of h_x, h_y. The target motion model is a constant velocity model, which is denoted by

$$X_k = AX_{k-1} + BW_k, \tag{1}$$

where A defines the deterministic component, B the stochastic component, and W_k is a multivariate Gaussian distribution.

The target color distribution is represented by color histogram. In our experiments, color histogram is calculated with m ($m = 8\times8\times8$) bins in R(red)G(green)B(blue) space. Assume that the target region (elliptic region) has a radius of $\mathbf{h} = (h_x, h_y)$ and is centered at $\mathbf{x} = (x, y)$, let $\mathbf{x}_i = (x_i, y_i)$, $i = 1, \cdots, n_h$, be the locations of the pixels in the target region, then the target color distribution is calculated as

$$\hat{p}^{(u)}(\mathbf{x}) = \sum_{i=1}^{n_h} k(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{\mathbf{h}} \right\|)\delta[b(\mathbf{x}_i)-u] \Big/ \sum_{i=1}^{n_h} k(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{\mathbf{h}} \right\|),$$

$$u = 1, \cdots, m, \tag{2}$$

where δ is the Kronecker delta function, $b(\mathbf{x}_i)$ is a function which associates the pixel at location $\mathbf{x}_i$ with the bin index $b(\mathbf{x}_i)$ of the histogram, $k(\cdot)$ is a weighting function (see Ref. [3]).

If denoting $\hat{q} = \{\hat{q}^{(u)}\}_{u=1,\cdots,m}$ as the color distribution of the target model and $\hat{p}(\mathbf{x}) = \{\hat{p}^{(u)}(\mathbf{x})\}_{u=1,\cdots,m}$ as a target candidate, the similarity between $\hat{q}$ and $\hat{p}(\mathbf{x})$ can be measured by

$$d_{\mathrm{c}}[\hat{q}, \hat{p}(\mathbf{x})] = \sqrt{1 - \rho[\hat{q}, \hat{p}(\mathbf{x})]}, \tag{3}$$

where $\rho[\hat{q}, \hat{p}(\mathbf{x})]$ is the Bhattacharyya coefficient that has the following form[3]

$$\rho[q, p(\mathbf{x})] = \sum_{u=1}^{m} \sqrt{\hat{q}^{(u)}\hat{p}^{(u)}(\mathbf{x})}. \tag{4}$$

After obtaining a distance d_{c} on the RGB color histogram, a color likelihood function is defined as

$$p_{\mathrm{c}}(Z_k|X_k) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{c}}} \exp\left\{-\frac{d_{\mathrm{c}}^2[\hat{q}, \hat{p}(X_k)]}{2\sigma_{\mathrm{c}}^2}\right\}, \tag{5}$$

where σ_{c} is the Gaussian variance, in our experiments, σ_{c} is selected as 0.3. Equation (5) shows that the

larger $p_c(Z_k|X_k)$ is, the more similarity between two histograms.

Because the target is represented by an ellipse, the target shape template is an ellipse. When measuring the similarity between two shapes, Chamfer distance is applied. Given a binary image T of the shape template, a binary image I_k, and the distance image DI_k of I_k, about the distance image, the reader can refer to Ref. [9], and assume the target candidate is at location $\mathbf{x} = (x, y)$ in the image I_k, then the Chamfer distance between the shape template and the target candidate is calculated as

$$d_s[T(\mathbf{x}), I_k] = \sqrt{\frac{1}{|T|}\sum_{t\in T} DI_k(t)}, \tag{6}$$

where $T(\mathbf{x})$ denotes the image T centered at $\mathbf{x} = (x, y)$ in the image I_k, $|T|$ denotes the number of features in the image T, and $DI_k(t)$ is the value of the pixel in the distance image DI_k which lies under the tth feature in the image T. After obtaining the Chamfer distance between two shapes, a shape likelihood function is defined as

$$p_s(Z_k|X_k) = \frac{1}{\sqrt{2\pi}\sigma_s}\exp\left\{-\frac{d_s^2[T(X_k), I_k)]}{2\sigma_s^2}\right\}, \tag{7}$$

where σ_s is the Gaussian variance, σ_s is set as 0.5 in our experiments. Equation (7) shows that the larger $p_s(Z_k|X_k)$ is, the more similarity between two shapes.

The observation model $p(Z_k|X_k)$ concerns in this paper two image cues: color cue and shape cue. Given the target state X_k at time k, the entire observation likelihood can be calculated as

$$p(Z_k|X_k) = \alpha p_c(Z_k|X_k) + \beta p_s(Z_k|X_k), \quad \alpha + \beta = 1, \tag{8}$$

where $p_c(Z_k|X_k)$ and $p_s(Z_k|X_k)$ are the likelihoods of the color and shape cues respectively, α, β $(0 \le \alpha, \beta \le 1)$ are the weights of the color and shape cues. The cue weights in most algorithms are assumed to be unchanged during the tracking. However, such assumption is often invalid in practice. Instead of assuming the fixed weights, fuzzy logic is used to adjust the weights dynamically.

Fuzzy logic consists of fuzzification, fuzzy rule base, fuzzy inference, and defuzzification[10]. In this paper, the fuzzy logic is designed based on the singleton fuzzification, product inference, and centroid defuzzification. Fuzzy logic inputs are the reliability levels of the color and shape cues in the current frame, while the output is the color weight α in the next frame. The shape weight β is calculated by Eq. (8). We denote the reliability levels of the color and shape cues by e_c and e_s respectively, which can be calculated by Eqs. (5) and (7). The fuzzy sets of e_c and e_s are {SR, S, M, B, BR} and α is {ST, VS, SR, S, M, B, BR, VB, BT}, where ST stands for smallest, VS for very smaller, SR for smaller, S for small, M for middle, B for big, BR for bigger, VB for very bigger, BT for biggest. The membership functions for e_c, e_s and α are all Gaussian functions, which are shown in Fig. 1. The fuzzy rule base is shown in Table 1. According to the fuzzy rules, the unreliable cues are given smaller weights while cues that have proved to be reliable are given larger weights.

The visual tracking problem can be formulated in a probabilistic framework by representing tracking as a

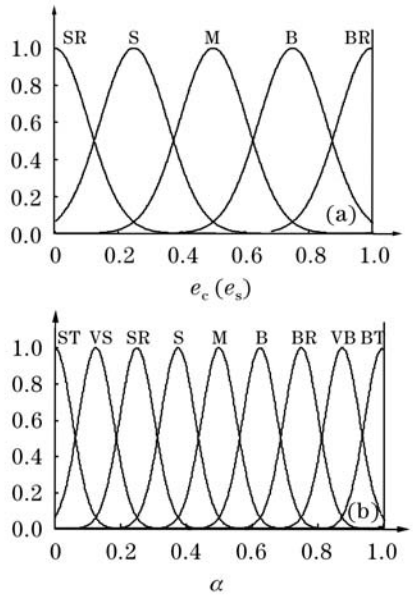

Fig. 1. Membership functions.

Table 1. Fuzzy Rule Base

α		e_c				
		SR	S	M	B	BR
e_s	SR	M	B	BR	VB	BT
	S	S	M	B	BR	VB
	M	SR	S	M	B	BR
	B	VS	SR	S	M	B
	BR	ST	VS	SR	S	M

process of posterior probability density propagation. Denoting the state of the target and the observation at time k by X_k and Z_k respectively, and denoting the state vectors and the observation vectors up to time k by $X_{0:k} = \{X_0, X_1, \cdots, X_k\}$ and $Z_{1:k} = \{Z_1, Z_2, \cdots, Z_k\}$ respectively, then in the Bayesian filtering framework[11], the tracking problem is formulated as

$$p(X_k|Z_{1:k}) = p(Z_k|X_k) \times \frac{\int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})\mathrm{d}X_{k-1}}{\int p(Z_k|X_k)p(X_k|Z_{1:k-1})\mathrm{d}X_k}, \tag{9}$$

where the likelihood $p(X_k|X_{k-1})$ expresses the motion model and $p(Z_k|X_k)$ is the observation model. Equation (9) provides an optimal solution for the tracking problem, but in visual tracking it is often non-linear and non-Gaussian, analytical solutions do not exist, leading to the use of Monte Carlo method.

Particle filter is a new filtering based on Monte Carlo method and can effectively deal with non-linear and non-Gaussian problems. The basic idea of particle filter is to approximate the posterior distribution $p(X_k|Z_{1:k})$ by a set of particles with associated weights $\{(X_k^{(n)}, \pi_k^{(n)})\}_{n=1,\cdots,N}$. Ideally, we would like the particles to be samples from the true posterior. However, in practice, the posterior probabilistic density function is not convenient or possible to sample from. An alternative

approach is to sample particles$\{X_k^{(n)}\}_{n=1,\cdots,N}$ from an importance function $q(X_k|X_{0:k-1}, Z_{1:k})$ and weight the particles according to the following importance weight

$$\pi_k^{(n)} = \pi_{k-1}^{(n)} \frac{p(Z_k|X_k^{(n)})p(X_k^{(n)}|X_{k-1}^{(n)})}{q(X_k|X_{0:k-1}^{(n)}, Z_{1:k})}, \quad \sum_{n=1}^{N} \pi_k^{(n)} = 1. \tag{10}$$

After weighting each particle, the particles are re-sampled according to their corresponding weights. Denoting the re-sampled particles by $\{\tilde{X}_k^{(n)}\}_{n=1,\cdots,N}$, then the posterior distribution $p(X_k|Z_{1:k})$ can be approximated as

$$p(X_k|Z_{1:k}) \approx \sum_{n=1}^{N} \pi_k^{(n)} \delta(X_k - \tilde{X}_k^{(n)}), \tag{11}$$

where δ is the Kronecker delta function. With the discrete approximation of $p(X_k|Z_{1:k})$, the output of the tracker can be obtained by Monte Carlo approximation of the expectation

$$\hat{X}_k = E(X_k|Z_{1:k}) \approx \frac{1}{N} \sum_{n=1}^{N} \tilde{X}_k^{(n)}. \tag{12}$$

The tracking process is as follows:

1) Initialization $k = 0$,

draw the states $X_0^{(i)}$ from the $p(X_0)$, $i = 1, \cdots, N$, and let $\alpha = \beta = 0.5$.

2) For $k = 1, 2, \cdots$

(a) Importance sampling step

Propagation: $X_k^{(i)} = AX_k^{(i)} + BW_k^{(i)}$, $i = 1, \cdots, N$,

calculating the color likelihood $p_c(Z_k|X_k^{(i)})$ according to Eq. (5),

calculating the shape likelihood $p_s(Z_k|X_k^{(i)})$ according to Eq. (7),

calculating the entire observation likelihood $p(Z_k|X_k^{(i)})$ according to Eq. (8),

evaluating the importance weights $\pi_k^{(i)}$ according to Eq. (10),

normalizing these weights $\tilde{\pi}_k^{(i)} = \pi_k^{(i)} / \sum_{j=1}^{N} \pi_k^{(i)}$.

(b) Re-sampling step

Calculating the cumulative probabilities C_k, $C_k^{(i)} = C_k^{(i-1)} + \tilde{\pi}_k^{(i)}$, $C_k^{(0)} = 0$, $i = 1, \cdots, N$;

for $i = 1, \cdots, N$,

generating a random number $r \in [0, 1]$, uniformly distributed,

finding, by binary subdivision, the smallest j for which $C_k^{(j)} \geq r$;

Setting $\tilde{X}_k^{(j)} = X_k^{(i)}$,

for $i = 1, \cdots, N$, setting $X_k^{(i)} = \tilde{X}_k^{(i)}$ and $\pi_k^{(i)} = 1/N$.

(c) Output step

$$\hat{X}_k = E(X_k|Z_{1:k}) \approx \frac{1}{N} \sum_{n=1}^{N} \tilde{X}_k^{(n)}.$$

(d) Calculating e_c and e_s according to Eqs. (5) and (7).

(e) Calculating α, β using fuzzy logic according to e_c and e_s.

Our method is evaluated on a sequence image with 500 frames[12]. This sequence simulates various tracking conditions, including illumination changes, pose variations, partial occlusions, cluttered backgrounds and camera motion. In the experiments, the target model is initialized manually. The number of particles is $N = 50$. We present the comparison between the results obtained by four kinds of methods. After conducting 100 Monte-Carlo runs, the results are shown in Figs. 2 and 3, and Table 2. Figures 2 and 3 give a one-time experimental result. Figure 2(a) shows tracking results based on color cue, the method is effective when target color has little change, but if target color changes dramatically, the target is lost (such as the 95th frame). Figure 2(b) shows the tracking results based on shape cue, the method can work well in simple background. However, when the background is cluttered (such as the 230th frame), the tracking fails. Figure 2(c) shows the tracking results

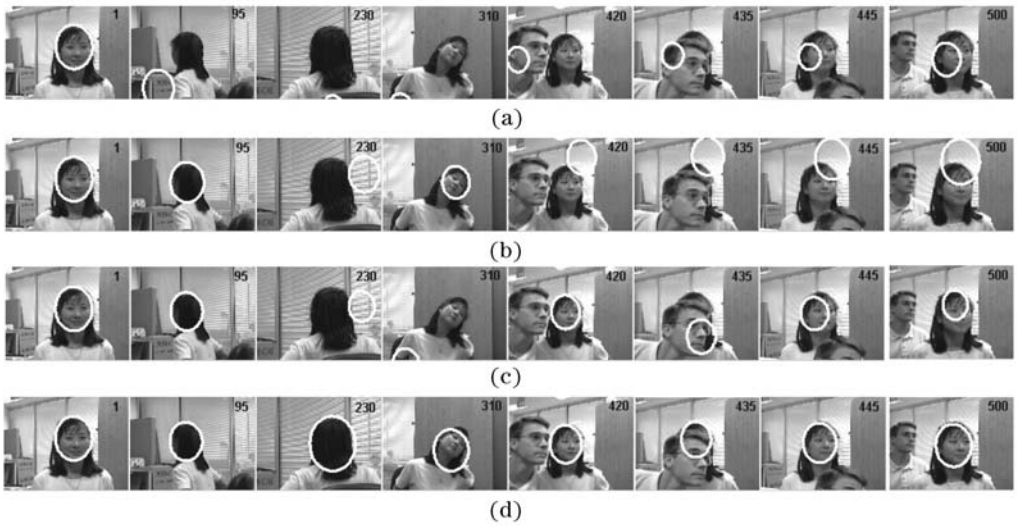

Fig. 2. Tracking results: (a) tracking based on color cue; (b) tracking based on shape cue; (c) tracking based on non-adaptive fusion; (d) our method.

using multiple cues, it can be seen that the results are better than Figs. 2(a) and (b). But the tracking still fails in the 230th frame. This is mainly because the reliability of each cue is assumed to be unchanged during the tracking, such assumption is invalid in our experiments due to the head rotation and the camera moving. Figure 2(d) gives the results of the proposed method. The tracking results show that the method can track the head robustly and accurately throughout the whole sequence image. Figure 3 gives weight curves of the color and shape cues, and shows that the proposed fusion scheme can successfully enhance the cue that is reliable for tracking. For example, at about the 65th frame, the color cue becomes unreliable due to the head rotation, while the shape cue can work well, so the color weight automatically decreases and the shape weight correspondingly increases to keep tracking robustly. Table 2 gives the stable root mean square (RMS) position error. From the Table 2, we can see that the tracking performance of our method is best.

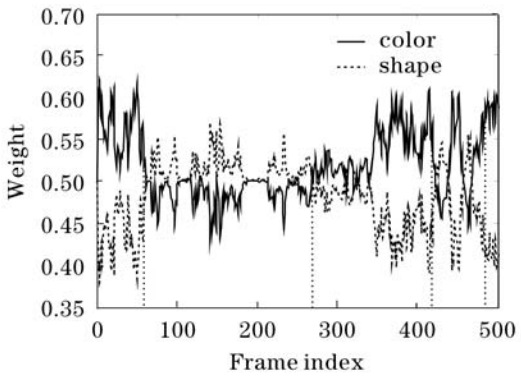

Fig. 3. Weight curves of the color and shape cues.

Table 2. Stable RMS Position Error

Methods	Stable RMS Position Error	
	x Direction (pixel)	y Direction (pixel)
Tracking Based on Color Cue	31.70	29.40
Tracking Based on Shape Cue	27.76	20.84
Tracking Based on Non-Adaptive Fusion	15.38	16.30
Our Method	6.53	7.93

A novel tracking method is proposed in this paper. The target observation is represented by multiple cues. When fusing each cue, an adaptive fusion scheme is applied. Particle filter is adopted to solve the non-linear and non-Gaussian problems in visual tracking. The experimental results show that with adaptive fusion, the tracker becomes more robust to illumination changes, pose variations, partial occlusions, cluttered backgrounds and camera motion.

In future work, more image cues will be considered to represent the target observation. When designing the particle filter, the important function is chosen as the transition prior, this choice works not well, so other choice method will be developed.

This work was jointly supported by the National Natural Science Foundation of China (No. 60375008), China PH.D Discipline Special Foundation (No. 20020248029), China Aviation Science Foundation (No. 02D57003), Aerospace Supporting Technology Foundation (No. 2003-1.3 02), EXPO Technologies Special Project of National Key Technologies R&D Programme (No. 2004BA908B07), and Shanghai Key Technologies Pre-research Project (No. 035115009). A. Li's e-mail address is lapjt@sjtu.edu.cn.

References

1. A. Blake, M. Isard, and D. Reynard, Artificial Intelligence **78,** 101 (1995).
2. M. Isard and A. Blake, in *Proceedings of the European Conference on Computer Vision* **1,** 343 (1996).
3. D. Comaniciu, V. Ramesh, and P. Meer, IEEE Trans. Pattern Analysis and Machine Intelligence **25,** 564 (2003).
4. Z. Liu and J. Yang, Chin. Opt. Lett. **2,** 390 (2004).
5. S. Birchfield, in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition* **232** (1998).
6. T. Xiong and C. Debrunner, in *Proceedings of the Conference on Analysis of Images and Patterns* **190** (2003).
7. B. Kwolek, in *Proceedings of the 8th European Conference on Computer Vision* **192** (2004).
8. M. Spengler and B. Schiele, Machine Vision and Applications **14,** 50 (2003).
9. D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, IEEE Trans. Pattern Analysis and Machine Intelligence **15,** 850 (1993).
10. L. X. Wang, IEEE Trans. Fuzzy System **1,** 146 (1993).
11. V. D. M. Rudolph, D. Arnaud, D. F. Nando, and W. Eric, CUED/F-INFENG/TR **380,** 1 (2000).
12. The image sequences are available at http://vision.stanford.edu/~birch/.