MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan
1-3

Fusing Color and Contour in Visual Tracking

Pham Nam Trung, Wu Jian Kang
Institue for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613

S.H. Ong
Department of Electrical and Computer Engineering,
National University of Singapore
Singapore 119260

Abstract

A tracking system with color and contour information is more efficient and robust than one with color or contour only. However, it is difficult to use both color and contour information. In this paper, we present an approach using the particle filter to fuse color and contour cues in tracking. First, we combine color and contour information in a Kalman filter to generate the proposal distribution which is one of the key points to improve the performance of particle filter. Subsequently, the particle filter will be applied to give the final tracking result. Our algorithm framework is flexible and it allows us integrate more measurements. Experimental result shows that this approach is an efficient and robust method.

1 Introduction

Object tracking is a core component technique in developing applications such as surveillance, smart room, and driver assistance. One of the main objectives of object tracking is to understand the motion of objects, track their trajectories and then analyze behaviors. There are many existing algorithms for tracking, such as mean shift [9], Kalman filter [1,4], and particle filter [1,2,3,5]. Normally, a visual tracking system uses a particular cue such as color [9] and contour information [6,7]. It is easy to see that along the path of the object, there will be many fluctuations in video capturing conditions, which will result in variations in object features such as color and contour shape. In this case, the performance of the tracking system will decrease if only one feature is used. A tracking system with both color and contour cues will have a number of advantages. Each feature may compensate for the weaknesses of the other. When the object color is changed, the performance can still be good because the contour feature may not be affected. On the other hand, in the case of a fast moving object when the contour is not reliable because of blurring at the object's boundary, we can use color information to track the object. However, there are many challenging issues that have to be solved to fuse multiple cues in tracking. For example, in order to fuse color and contour information, the fusion system must be able to decide when and which cue is the most suitable and to what degree.

Due to the many advantages of integrating features in tracking, there are some publications about multiple cues in tracking in recent years. Perez et al. (2004) introduced generic mechanisms for data fusion using a particle filter based on layer sampling [12]. This method searches sequentially in each direction rather than the entire space. So, it is efficient in the case that state space can be partitioned for searching. Chen (2003) proposed a method using a mixture of each distribution from the tracking modules to form a fused proposal in particle filter [10]. However, in many cases, it would be difficult to define the coefficient parameters for each component. Spengler and Schiele (2003) integrated multiple cues and propose self adaptation of each cue during tracking [14]. G. Taylor and L. Kleeman (2003) used Kalman filter to fuse visual cues in their system [16].

In this paper, we propose a framework to fuse multiple cues using the particle filter. One of the key points to improve the performance of the particle filter is to generate a good proposal distribution. This leads to the idea of combining cues in the Kalman filter and using it as proposal distribution. After that we employ the particle filter to do the tracking. Here, we use color and contour features in the tracking system. Mean shift color tracking [9] and a simple search edge contour fitting will be applied to obtain real observations for the Kalman filter.

The proposed method here provides a mechanism for fusion of multiple cues in tracking. If the contour information is not suitable for tracking, color information from mean shift tracking will be able to correct the predicted state in the Kalman filter. Then, the result of Kalman filter tracking will be a good proposal distribution for the particle filter. Conversely, if the color information is not good enough for tracking, the predicted state of the object in the Kalman filter will be corrected by contour observation via the greedy search algorithm.

The rest of the paper is organized as follows. Section 2 describes the tracking algorithm. Section 3 gives an overview of color and contour features. Finally, section 4 provides implementation details and experimental results to demonstrate the efficiency and robustness of the method.

2 Tracking Algorithm Framework

2.1 Fusion multiple cues in the Kalman filter

Our tracking algorithm uses the Kalman filter to derive the proposal distribution for the particle filter from the idea of P. Li [7]. In 1960, Kalman (1960) published his famous paper describing a recursive solution to the discrete data linear filtering problem. The Kalman filter is used with a linear Gaussian and Markov assumption. The model of a linear dynamic process for the Kalman filter is defined by

$$x_t = F_t x_{t-1} + w_t, \quad w_t \sim N(0, Q_t) \qquad (1)$$

where the system transition matrix F_t models the evolution of the state vector x_t. The measurement model

$$z_t = H_t x_t + v_t, \quad v_t \sim N(0, R_t) \qquad (2)$$

determines the measurement z_t as a function of the state x_t. H_t is called the measurement sensitivity matrix. The system noise w_t and the measurement noise v_t are zero mean Gaussian sequences with given covariance matrixes Q_t and R_t, respectively.

For fusing multiple cues, z_t multiple measurements are extracted from image, so the dimensionsionality in z_t depends on the number of features used in the tracking system. The basic steps of the computational procedure for Kalman filter with multiple cues are as algorithm 1.

$\left(\bar{x}_t^{(i)}, P_t^i\right) = KalmanFilterFusion\left(x_{t-1}^{(i)}, P_{t-1}^i\right)$

1. A priori state estimate extrapolation
   $\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1}$
2. Error covariance extrapolation
   $P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t$
3. Kalman gain matrix
   $K_t = P_{t|t-1} H_t^T [H_t P_{t|t-1} H_t^T + R_t]^{-1}$
4. Error covariance update
   $P_{t|t} = [I - K_t H_t] P_{t|t-1}$
5. A posteriori state estimate observational update
   $\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t [z_t - H_t \hat{x}_{t|t-1}]$, where z_t stored multiple measurements extracted from image.

Algorithm 1. Fusion of multiple cues in the Kalman filter

2.2 Fusion multiple cues in particle filter

The particle filter has been developed to solve non-Gaussian and non-linear problems. The posterior density $p(x_t \mid z_{1:t})$ in the particle filter is approximated by a set of discrete samples with associated weights $\left\{x_t^{(i)}, w_t^{(i)}\right\}$. The algorithm involves three steps. The first step is the sampling step. Particles $x_t^{(i)}$ are drawn from a proposal distribution $q\left(x_t \mid x_t^{(i)}, z_t\right)$ and are reweighted by their likelihood according to formula

$$w_t^{(i)} = w_{k-1}^{(i)} \frac{p(z_t \mid x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)})}{q(x_t^{(i)} \mid x_{t-1}^{(i)}, z_t)} \tag{3}$$

The sampling technique using in the particle filter is important sampling method. In the second step, particles are resampled to form a uniform weight distribution. Finally, the filter outputs the estimates of the state, such as the mean and covariance [1, 2].

One problem with the particle filter is that if the proposal sampling function is not near with the posterior, there are only a few particles dominating the particle cloud and most of them will be removed because of low probability. As such, it is important to have a good proposal distribution. In this paper, we use the Kalman filter introduced in 2.1 to generate the sampling function. We also assume that multiple cues are independent, so the likelihood can be calculated by the formula

$$p(z_t \mid x_t^{(i)}) = \prod_{k=1}^{K} p(z_t^k \mid x_t^{(i)}),$$

where K is number of cues.

The steps of the computational procedure for the Kalman particle filer with multiple cues are set out in algorithm 2.

Initialization (t=0)
For i = 1,...,N
Set $w_0^{(i)} = 1/N$ and draw $x_0^{(i)} \sim p(x_0)$
For t = 1, 2,...
Importance Sampling step
For i = 1,...,N
On each $x_{t-1}^{(i)}$ apply Kalman Filter to obtain $\bar{x}_t^{(i)}, P_t^i$
$\left(\bar{x}_t^{(i)}, P_t^i\right) = KalmanFilterFusion\left(x_{t-1}^{(i)}, P_{t-1}^i\right)$
Draw $\hat{x} \sim q(x_t \mid x_{t-1}^{(i)}, z_t) = N(\bar{x}_t^{(i)}, P_t^i)$
Evaluate Likelihood

$$p(z_t \mid x_t^{(i)}) = \prod_{k=1}^{K} p(z_t^k \mid x_t^{(i)})$$

Evaluate (unnormalized) importance weights

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(z_t \mid x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)})}{q(x_t^{(i)} \mid x_{t-1}^{(i)}, z_t)}$$

For i = 1,...,N
Normalized the importance weights

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}}$$

Resampling step
Obtain new samples $\left\{x_t^{(i)}\right\}_{i=1}^{N}$ by sampling N times with replacement from $\left\{\hat{x}_t^{(i)}\right\}_{i=1}^{N}$ such that $\Pr\left\{x_t^{(i)} = \hat{x}_t^{(j)}\right\} = w_t^{(j)}$
Reset $w_t^i = 1/N$
Output step

$$\hat{x}_{t|t} = \sum_{i=1}^{N} w_t^{(i)} x_t^{(i)}$$

End

Algorithm 2. Fusion of multiple cues in the Kalman particle filter

3 Color and contour feature in Kalman particle filter

3.1 Color likelihood

We use the method proposed by Comaniciu [9] to compute color likelihood. Let us denote the color histogram distribution of an candidate object by *p(u)*, and the color histogram distribution of the template by *q(u)*. The similarity function between the candidate object and the template is measured by the Bhattacharyya distance. First, we compute the Bhattacharyya coefficient.

$$\rho[p,q] = \int \sqrt{p(u) q(u)} du \tag{4}$$

The distance between two distributions

$$d = \sqrt{1 - \rho[p,q]}, \tag{5}$$

is called the Bhattacharyya distance. Color likelihood is computed as the Gaussian distribution

$$p(z_t^{color} \mid x_t^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{d^2}{2\sigma^2}} \quad (6)$$

3.2 Contour likelihood

MacCormick (2000) introduced a method to compute contour likelihood [11]. Contour observation is based on edge measurements. Edge-based measurements are performed in 1D along normal lines to a hypothesized contour. In this paper, we use the ellipse hypothesized contour. Denoting the scalar positions in k^{th} normal line by $(v_k^1,\ldots,v_k^{m_k})$ (m_k is the number of detected edge points in the normal line k^{th}), we have

$$p(z_t \mid x_t^i, v_k) = \frac{e^{-\lambda L}\lambda^{m_i}}{m_i!}\left(q_{01} + \frac{q_{11}}{\lambda}\sum_{l=1}^{m_i}\exp\left(-\frac{(z_t^k - v_k^l)^2}{2\sigma^2}\right)\right) \quad (7)$$

where q_{01} is the probability of a missing edge point, q_{11} is the probability of a detected edge point and λ is the mean of a number of detected points in a normal line. The contour likelihood function is

$$p(z_t \mid x_t^i) = \prod_{k=1}^{K} p(z_t \mid x_t^i, v_k), \quad (8)$$

where K is the number of normal lines.

3.3 Contour and color observation in Kalman filter

Comaniciu introduced mean shift color tracking in [9]. This method is a local minimum optimization tracking method. Each location x_i in the candidate region of the tracking will be associated with a weight

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}\,\delta[b(x_i) - u], \quad (9)$$

where $b(x_i)$ is the color value at x_i, $\hat{q}_u$ is the value at color u of the target model, and $\hat{p}_u(\hat{y}_0)$ is the value at color u of the candidate model. $\hat{q}, \hat{p}$ are computed by

$$\hat{q}_u = C\sum_{i=1}^{N} k\left(\|x_i^*\|^2\right)\delta[b(x_i) - u], \quad (10)$$

$$\hat{p}_u(y) = C_h\sum_{i=1}^{N} k\left(\left\|\frac{y - x_i^*}{h}\right\|^2\right)\delta[b(x_i) - u] \quad (11)$$

where δ is the Kronecker delta function and k is the Epanecnikov kernel function.

The new object location is computed by

$$\hat{y}_1 = \frac{\sum_i x_i w_i}{\sum_i w_i} \quad (12)$$

The process is repeated until there is no change in the new location. We obtain the estimated location from mean shift tracking, and the major and minor ellipse radius estimated in the previous time as the color observation in the Kalman filter.

For contour observation in Kalman filter, we implemented a greedy algorithm to obtain ellipse fitting with edge points. The details of this are shown in algorithm 3.

```
(x̂, ŷ) = FindContour(x, y, rx, ry)
while ((x,y) not change)
    for each (xi, yi) is neighbor with (x,y)
        Compute number of edge points near with
        ellipse (xi, yi, rx, ry)
    (x,y) = ellipse that have maximum number of edge
    points near with it.
End
(x̂, ŷ) = (x, y)
```

Algorithm 3. Finding ellipse contour observation in the Kalman filter

4 Experimental results

We tested our algorithm on two sequences of images. The first sequence is from http://vision.stanford.edu/~birch/headtracker/ and there are 500 frames in the sequence. In this sequence, the color of the girl's head changes when moving. The results of tracking in this sequence is shown in figure 1 and 2 for the generic particle filter based color approach and our proposed approach. In the second sequence, there are 41 frames. In this sequence, when the ball moves fast, it is difficult to do edge detection of the ball. The result of tracking in this sequence is shown in figure 3 and 4 for the generic particle filter based contour approach and our approach.

We use the constant velocity model for dynamic moving equation

$x_t = x_{t-1} + w_t, \quad w_t \sim N(0, Q_t)$

The state of object x_t is a 4-dimensional vector that includes the center point and the major and minor radius of the ellipse to describe the object. The variance in color likelihood is $\sigma_1 = 0.005$, the number of normal lines m = 10, the length of normalized L = 12, parameters $\lambda = 0.3$, and the variance for contour likelihood is $\sigma_2 = 2$.

It is easy to see that in the two sequences when features change, the generic particle filter cannot track the object, but our approach still has a good performance.

5 Conclusion

The article proposed a robust way to integrate multiple cues in particle filter. With this approach, we could add more features to improve tracking performance. Experimental results have shown that this proposed approach is quite robust and flexible.

References

[1] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for on-line non-linear/non gaussian bayesian tracking", *IEEE Transactions on Signal Processing, Special Issue on Monte Carlo Methods*, 50(2), pp 174-188; February 2002.

[2] A. Doucet, N. Freitas, N. Gordon, 'Sequential monte carlo methods in practice', Springer-Verlag, 2001.

[3] R. V. Merwe, A. Doucet, N. Freitas, E. Wan, "The unscented particle filter", *Technical report CUED/F_INPENG/TR 380,*

Cambridge University Engineering Department, 2000.

[4] B. Anderson, J. .Moore, ‘Optimal filtering’, Prentice Hall, New Jersey, 1979.

[5] M. Isard and A. Blake, “Condensation: conditional density propagation for visual tracking”, *International J. Computer Vision*, 28(1), 1998.

[6] A. Blake and M. Isard, ‘Active contours’, Springer-Verlag, 1998.

[7] P. Li, T. Zhang, A. Pece, “Visual contour tracking based on particle filters”, *Image and Vision Computing 21*, page 111-123, 2003.

[8] Y. Rui and Y. Chen, “Better proposal distributions: object tracking using unscented particle filter”, *Proc. of IEEE CVPR 2001* , pp. II-786 to 793, Kauai, Hawaii, December 11-13, 2001.

[9] D. Comaniciu, V. Ramesh, P. Meer, “Kernel based object tracking”, *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 25, No. 5, 564-575, 2003.

[10] Y. Chen, “Robust object tracking based on multiple cues”, PhD Thesis, University of Illinois at Urbana-Champaign, 2003.

[11] J. MacCormick, “Probabilistic modelling and stochastic algorithms for visual localisation and tracking”, PhD Thesis, University of Oxford, 2000.

[12] P. Perez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particles”, *Proceedings of the IEEE*, March 2004.

[13] S. Zhou, R. Chellappa, B..Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters", *IEEE Transactions on Image Processing (TIP)*, To Appear December 2004.

[14] M. Spengler, B. Schiele, “Towards robust multi-cue integration for visual tracking”, *Machine Vision and Applications*, 2003.

[15] C. Shen, A. Hengel, A. Dick, “Probabilistic multiple cue integration for particle filter based tracking”, *Proc. VIIth Digital Image Computing: Techniques and Applications*, Sydney, 2003.

[16] G. Taylor, L. Kleeman, “Fusion of multimodal visual cues for model-based object tracking”, *Proceedings of Australasian Conference on Robotics & Automation*, December 2003.

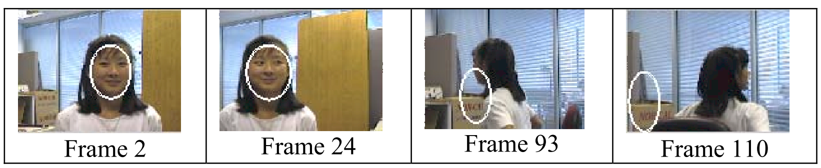

Figure 1. Head tracking with color observation with the generic particle filter

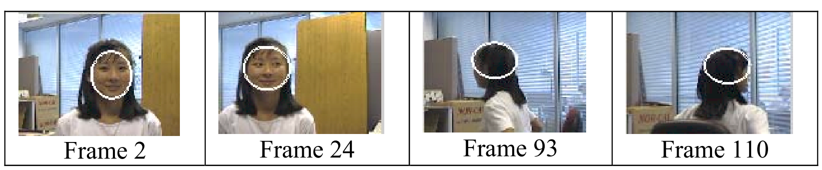

Figure 2. Head tracking with fusion color and contour with the Kalman particle filter

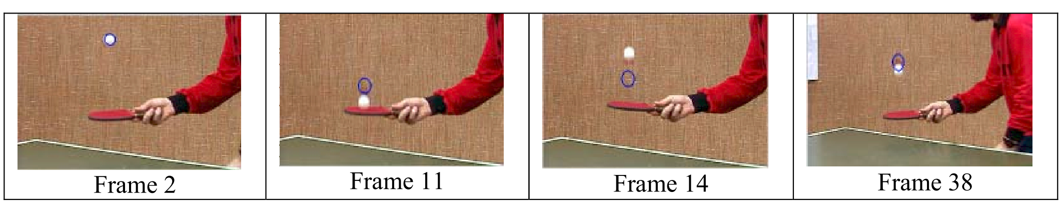

Figure 3. Ball tracking with contour observation in the generic particle filter

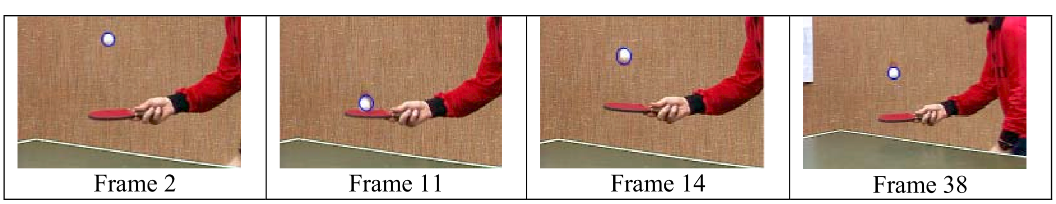

Figure 4. Ball tracking with fusion color and contour with the Kalman particle filter