Bayesian Filtering and Integral Image for Visual Tracking

Bohyung Han Changjiang Yang Ramani Duraiswami[†] Larry Davis

{bhhan, yangcj, lsd}@cs.umd.edu †ramani@umiacs.umd.edu

Dept. of Computer Science, University of Maryland, College Park MD 20742, USA

Abstract

This paper describes contributions to two problems related to visual tracking: control model design and observation process design. We describe the use of kernel-based Bayesian filtering for the tracking control procedure, and feature-based tracking to improve the observation process of tracking. In the kernel-based Bayesian filtering framework, the analytical representation of density functions by density interpolation and density approximation for the likelihood and the posterior contributes to efficient sampling. Feature-based tracking combines simple rectangular features with edge oriented histogram so that the combined features are robust to illumination changes, partial occlusion, and clutter while capturing the spatial information of the target. The use of integral image allows the features to be efficiently evaluated. The effectiveness of both algorithms are demonstrated by object tracking results on real videos.

1 Introduction

Tracking algorithms generally involve addressing two basic problems – tracking control and observation. A significant body of research addresses these two issues – for example, [3, 5, 7, 13].

Based on the nature of their control procedures, tracking algorithm can be classified into two categries: deterministic and stochastic methods. Deterministic methods typically track by performing an iterative search for the local maximum (or minimum) of a similarity cost function between the target and the candidates. In stochastic methods, tracking is performed by estimating the probability density function in the state space of target motion and transformations. Deterministic trackers are fast but sensitive to occlusion and clutter, while stochastic ones are more robust and capable of recovering from temporary tracking failures.

The target representation and similarity measurement – observation model – are also very important to the performance of trackers, and their design is closely related to the feature selection problem. The color histogram is widely used in object tracking algorithms since it is robust to noise and partial occlusion. However, it cannot deal easily with illumination changes and lacks any spatial information. On the other hand, contour-based methods are more invariant to lighting conditions, but computationally expensive and not robust to clutter.

In this paper, we describe our research on the tracking control and observation problems. In section 2, a Bayesian filtering framework based on a density approximation technique is presented, where the probability density function in each step is a mixture of Gaussians. The advantage of maintaining an analytic representation of density functions lies in efficient sampling, and the density interpolation is incorporated for accurate approximation of the measurement function. The feature combination of the Harr-like rectangular features [14] and the edge orientation histogram [5] as an observation model is discussed in section 3. The combined features are evaluated efficiently and used in the measurement step to improve the speed and robustness of the observation.

2 Tracking by Bayesian Filtering

In this section, we introduce a Bayesian filtering framework, where the relevant density functions are approximated by kernel-based representations and propagated over time. This work is an extension of a previous paper [6], which includes more mathematical details.

2.1 Kernel-Based Bayesian Filtering

In a dynamic system, the process and measurement model are given by

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) \tag{1}$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t) \tag{2}$$

where $\mathbf{v}_t$ and $\mathbf{u}_t$ are the process and the measurement noise, respectively. In the sequential Bayesian filtering framework, the conditional density of the state variable given the prior $p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})$ and the measurement $p(\mathbf{z}_t|\mathbf{x}_t)$ is propagated through prediction and update stages as,

$$p(\mathbf{x}_t|\mathbf{z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})d\mathbf{x}_{t-1} \tag{3}$$

$$p(\mathbf{x}_t|\mathbf{z}_t) = \frac{1}{k}p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{t-1}) \tag{4}$$

where $k = \int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{t-1})d\mathbf{x}_t$ is a normalization constant. The posterior at time step t $p(\mathbf{x}_t|\mathbf{z}_t)$ is used as the prior in step $t+1$.

In our kernel-based Bayesian filtering, the posterior density function is represented with a Gaussian mixture whose number of components is equal to the number of modes in the underlying density. Therefore, our main goal is to retain such a representation through the prediction and update steps, and to represent the posterior density in the following step with a Gaussian mixture form.

The details of the proposed filtering framework are described next. First, the unscented transformation (UT) [8, 12] is used in the prediction step to maintain a Gaussian mixture representation even with non-linear process model. By the UT, the predicted density function is accurate up to the second order of the Taylor expansion. Second, samples are drawn from the proposal distribution – $p(\mathbf{x}_t|\mathbf{z}_{t-1})$, and the measurement density function is derived by density interpolation technique whose output is also a Gaussian mixture. The posterior is obtained by multiplying two mixtures, prediction and measurement functions. To prevent the number of mixands from growing too large, density approximation technique [6] based on variable-bandwidth mean-shift is applied to derive a compact representation for the posterior density. After the update step, the final posterior density is given by a compact Gaussian mixture form.

The most important characteristic of our Bayesian filtering is that every density function is represented with a continuous function – a Gaussian mixture. The analytical representation of density functions and the kernel-based particles contribute to efficient sampling by which the number of samples can be reduced significantly in high dimensional problems. Also, this technique can be applied to the cases in which the only small number of samples is available or the measurement for each particle is very expensive.

2.2 Density Interpolation

In the measurement step, the likelihood values are known for a set of samples, and the measurement density can be interpolated with sample likelihoods.

A Gaussian kernel is assigned to each sample whose mean and covariance corresponds to the sample location and the bandwidth, respectively. The bandwidth is selected based on k-nearest neighbors (KNN), which is similar to the methods discussed in [1, 2]. In short, each sample is intended to cover the local region around itself in the d-dimensional state space with its bandwidth, and the kernel bandwidth is determined by the distance to the k-th nearest neighbor of the sample. By this method, samples in dense areas have small bandwidths and the density will be represented accurately, but sparse areas convey only relatively rough information about the density function.

Denote $\mathbf{x}_i$ as the mean location and $\mathbf{P}_i$ as the covariance matrix for the i-th sample. Also, suppose that l_i is the likelihood value on the i-th sample. Given $\mathbf{x}_i$, $\mathbf{P}_i$ and l_i ($i = 1, \ldots, n$), the non-negative least square (NNLS) method [9] is employed to compute the weight as follows. Define an $n \times n$ matrix $\mathbf{A}$ having a probability of j-th sample w.r.t. i-th kernel with a unit weight in (i, j), and an $n \times 1$ vector $\mathbf{b}$ having l_i in its i-th row. Then, the weight vector $\mathbf{w}$ can be computed by solving the following constrained least square problem,

$$\min_{\mathbf{w}} ||\mathbf{A}\mathbf{w} - \mathbf{b}||^2 \tag{5}$$

$$\text{subject to } \mathbf{w}_i \geq 0 \text{ for } i = 1, \ldots, n.$$

The size of matrix $\mathbf{A}$ is determined by the number of samples. When the sample size is large, sparse matrix operation methods can be used to solve $\mathbf{w}$ efficiently.

Usually, many of the weights will be zero and the final density function will be a mixture of Gaussians with a small number of components. The density interpolation simulates the heavy-tailed density function more accurately than the density approximation introduced in [6], while the density approximation generally produces a more compact representation.

2.3 Object Tracking

Our kernel-based Bayesian filtering has the advantage of managing multi-modal density functions with a relatively small number of samples. In this section, we demonstrate the performance of the kernel-based Bayesian filtering by tracking objects in real videos.

The basic Bayesian filtering steps are not changed, and the process and the measurement models are described below.

- **process model**: random walk ($\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{u}_t$)
- **measurement model**: inverse exponentiation of Bhattacharyya distance between the target and the candidate histograms as suggested in [13]

In our experiment, two different sequences are tested. In the first sequence, two objects – a hand carrying a can – are tracked with 200 samples. The state space is described by a 10 dimensional vector, which is the concatenation of two 5 dimensional vectors representing two independent ellipses. The tracking result is shown in figure 1, and our algorithm successfully tracks two objects.

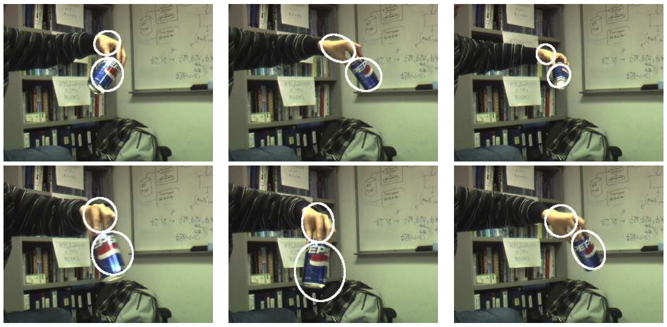

Figure 1: Object tracking result of *can* sequence at $t = 47, 112, 187, 278, 360, 400$.

The upper bodies of two persons are tracked in the second sequence, in which one occludes the other completely several times. A 6 dimensional vector – $(x, y, scale)$ for each rectangle – is used to describe the state, and 100 samples are used. Figure 2 demonstrates the tracking results, and our algorithm shows good performance in spite of severe occlusions.

For comparison, the tracker based on SIR particle filter is also implemented, and compared with our algorithm. The SIR algorithm shows unstable performance for the same sequence, and, according to experiments, one would need to run the SIR algorithm using at least 400 particles to obtain a comparable result with our algorithm using 100 samples.

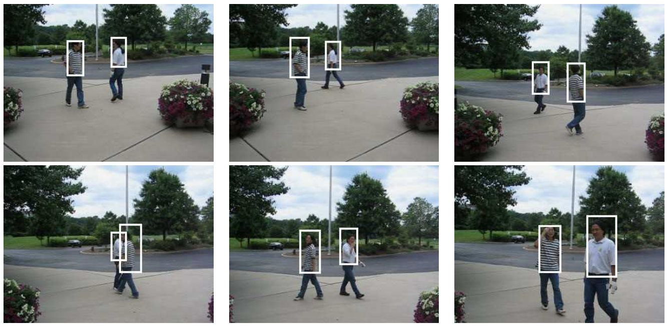

Figure 2: Object tracking result of *person* sequence at $t = 1, 92, 134, 193, 216, 300$.

3 Feature-Based Tracking

The observation model is used to measure the observation likelihood of the samples. Many observation models have been built for particle filtering tracking. The tracker based on a contour template [7] gives an accurate description of the targets but performs poorly in clutter and is generally time-consuming. The initialization of the system is relatively difficult and tedious. In contrast, color-based trackers are faster and more robust, where the color histogram is typically used to model the targets to combat the partial occlusion, and non-rigidity [13, 3]. The drawback of the color histogram is that spatial layout is ignored, and the trackers based on it are easily confused by a background with similar colors. The combination of the two features provides better performance which is described in the follows.

3.1 Color Rectangle Features

The rectangle features were introduced by Viola and Jones for real-time object detection [14]. In their method, the grayscale image was converted to integral image format (an image in which at each pixel the value is the sum of all pixels above and to the left of the current position). The sum of the pixels within any rectangle can then be computed in four table lookup operations on the integral image. The color images can be treated as multi-channel intensity images to generate multi-channel integral images.

To model the target using color information, we pick n rectangular regions $R_1, \dots, R_n$ within the object to be tracked. Each rectangle R_i is represented by the mean (r, g, b) color of the pixels within region R_i (other color spaces can be considered similarly)

$$(r_i, g_i, b_i) = \sum_{(x,y)\in R_i} \left(r(x,y), g(x,y), b(x,y)\right) / A_i, \qquad (6)$$

where A_i is the number of pixels within R_i. The mean color vector $(r_i^\star, g_i^\star, b_i^\star)$ of each region R_i can be computed during initialization. The reason we choose this color representation instead of the popular color histogram is that it encodes the spatial layout of the targets and offers robustness against noise. Such a color representation of the targets has been used in [4] for head tracking, where a hypothesize-and-test procedure is used to find a match between frames. In [4], for real time performance they can only consider a relatively small number of hypotheses, since there was no efficient way to evaluate the rectangle features.

The similarity between the template and the current frame $\rho(\mathbf{k}^\star, \mathbf{k}(\mathbf{x}_t))$ can be measured by the Euclidean distance between these rectangle features. The likelihood distribution is given by $p(\mathbf{z}_t|\mathbf{x}_t) \propto e^{-\rho^2(\mathbf{k}^\star, \mathbf{k}(\mathbf{x}_t))/\sigma^2}$.

The number of rectangles within the object can be as small as two in the first stage. There are two reasons for such a setup: one is efficiency, the other is that we want to keep more candidates in the first stage for robustness and prune those mostly negative samples as soon as possible. This strategy has been proven successful in object detection [14], and we observe the same for tracking.

3.2 Edge Orientation Histogram

To detect edges, we first convert color images to grayscale intensity images. Edges are detected using the horizontal and vertical Sobel operators: K_x and K_y:

$$G_x(x,y) = K_x * I(x,y), \quad G_y(x,y) = K_y * I(x,y). \qquad (7)$$

The strength and the orientation of the edges are

$$\begin{aligned} S(x,y) &= \sqrt{G_x^2(x,y) + G_y^2(x,y)}, & (8)\\ \theta &= \arctan\left(G_y(x,y)/G_x(x,y)\right). & (9) \end{aligned}$$

We also apply a threshold T to $S(x,y)$ to remove noise (T is set to 100 in our experiments). The edges are counted into K bins with their strengths $S(x,y)$. Figure 3 shows an example of the global edge orientation histogram of the walker in the image.

The edge orientation histogram within a rectangle region can be efficiently computed by treating it as K separated channels and accumulating K integral images [10]. The i-th bin value within a rectangle is the sum computed by four table lookup operations on i-th integral image.

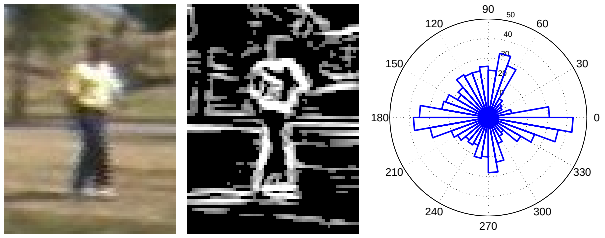

Figure 3: Edge orientation histogram. (*Left*) Example image. (*Center*) Edge strength image. (*Right*) Polar plot of edge orientation histogram.

The similarity between the template and the current image is computed with the Euclidean distance between the two global edge orientation histograms as in [11].

3.3 Experiments

We use the Harr-like features to simultaneously track multiple objects. Each object is associated with a individual template and 1000 particles. Each object moves independently. The proposed algorithm successfully tracked all objects through all frames. The image size is 352×288. Examples of the tracking results are shown in Figure 4. The tracking time with respect to the frame index is shown in the left panel of Figure 5. The average tracking time for single object with respect to the number of particles is shown in the right panel of Figure 5. The same procedures and configurations are applied to the color histogram based tracker and the corresponding tracking time is shown in Figure 5. We find that the once the integral images are built, the evaluation of the observation likelihood by the proposed method is independent of the size of regions and is very efficient. In contrast, for the color histogram based method or other methods, the bottle-neck is the building of the histograms whose complexity is proportional to the number of particles and the size of the regions.

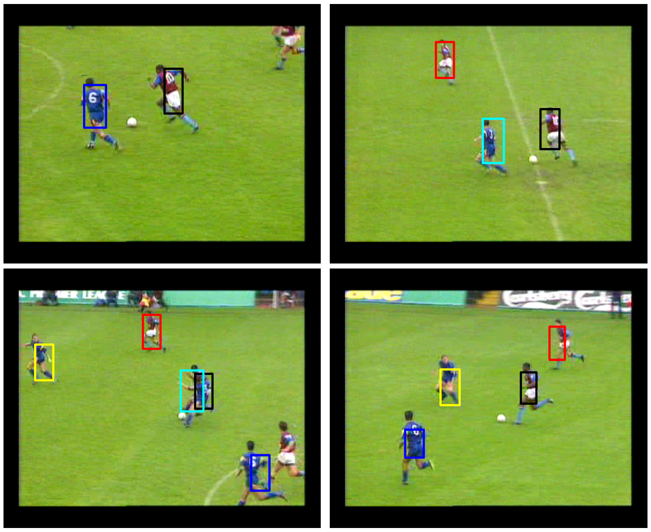

Figure 4: Results of the proposed particle filter based multiple object tracking for the football game sequence.

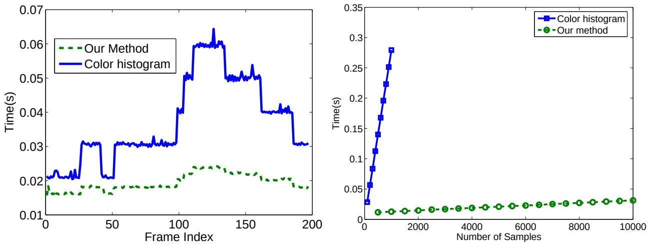

Figure 5: (*Left*) Tracking time w.r.t. the frame index. (*Right*) Tracking time w.r.t. the number of samples.

References

[1] W. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assn.*, 74:829–836, 1979.

[2] W. Cleveland and C. Loader. Smoothing by local regression: Principles and methods. *Statistical Theory and Computational Aspects of Smoothing*, pages 10–49, 1996.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5):564–577, 2003.

[4] P. Fieguth and D. Teropoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pages 21–27, 1997.

[5] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995.

[6] B. Han, D. Comaniciu, Y. Zhu, and L. Davis. Incremental kernel density approximation and kernel-based bayesian filtering for object tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington DC, volume I, pages 638–644, June 2004.

[7] M. Isard and A. Blake. Condensation - Conditional density propagation for visual tracking. *Intl. J. of Computer Vision*, 29(1), 1998.

[8] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings SPIE*, volume 3068, pages 182–193, 1997.

[9] C. L. Lauwon and B. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[10] K. Levi and Y. Weiss. Learning object detection from a small number of example: the importance of good features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington DC, pages 53–60, 2004.

[11] D. G. Lowe. Distinctive image features from scal-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004.

[12] R. Merwe, A. Doucet, N. Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.

[13] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proc. European Conf. on Computer Vision*, Copenhagen, Denmark, volume I, pages 661–675, 2002.

[14] P. Viola and M. Jones. Robust real-time face detection. *Intl. J. of Computer Vision*, 52(2):137–154, 2004.