HEAD TRACKING USING PARTICLE FILTER WITH INTENSITY GRADIENT AND COLOR HISTOGRAM

Xinyu Xu and *Baoxin Li*
Center for Cognitive Ubiquitous Computing
Dept. of Computer Science & Engineering
Arizona Statue University
Tempe, AZ, U.S.A.
{*xinyu.xu, baoxin.li*}*@asu.edu*

ABSTRACT

This paper presents a method for tracking human head using a particle filter to naturally integrate two complementary cues: intensity gradient and color histogram. The shape of the head is modeled as an ellipse, along which an intensity gradient is estimated, while the interior appearance is modeled using a color histogram. These two cues play complementary roles in tracking a human head with free rotation on a cluttered background. To evaluate the tracker performance, we test with both synthetic image sequences and real sequences. Experiments show that the tracker is robust to 360-degree rotation of the head on a cluttered background.

1. INTRODUCTION

Head tracking can find a lot of applications including automated surveillance, video conference, etc. However, developing a head tracking algorithm that is robust to a wide variety of conditions remains to be a challenging problem. Part of the challenge comes from the lack of reliable cues for tracking in all applications. For example, color-based trackers may be distracted by other targets of similar color, while edge-based trackers can be misled by clutters in the background[1]. In addition, compared with face tracking (e.g.,[7]), typically a head tracker must handle a larger degree of head rotation, which makes the color cue fragile under large head motion, as the color of the front of head (face) is typically totally different from that of the back of the head. Thus, combining multiple cues may be the only option.

Recent work on tracking human head includes those employing 3-D model for head pose estimation (e.g. [2,3]) and those using both color and shape cues (e.g. [5]). In this paper, we propose a method that uses the particle filter to integrate both color (in terms of color histogram) and shape (in terms of intensity gradient along the head contour) information in order to handle the above challenges. Both of the cues are based on 2-D models, hence allowing efficient implementation. Furthermore, using a particle-filter-based tracking framework provides desired robustness and accuracy as no assumption on linearity and Gaussian density are required to make about the problem.

In addition to using a particle filter framework, the proposed method is different from the approach of [5] in that: (1) the gradient is not simply estimated in the normal direction of an ellipse model; instead, a local search is incorporated for accounting for the inaccuracy of the ellipse model; (2) unlike in [5], the color model is adaptive in this paper.

In Sect. 2, we describe the proposed method, and then we present comparative experiments in Sect. 3, with both synthetic and real sequences. We conclude with Sect. 4.

2. PROPOSED METHOD

We first describe the general framework of the proposed tracker in Sect. 2.1, and then we present the details of the algorithmic components in Sect.2.2 through Sect.2.4.

2.1. A Head Tracking Framework Based on Particle Filter

Particle filter has been used in various tracking problems (e.g., [1][4]). In this work, we adopt a framework similar to that used in [4]. Specifically, the head is modeled as an ellipse centered at (x,y) and with size (H_x, H_y). At the first frame, we use the face detection algorithm proposed by [6] to detect the location and size of the head ellipse (if the face is frontal; otherwise, a manual initialization is performed by clicking on the head). The dynamics of the (moving) head at time t are described by a state vector S_t consisting of the following eight components

$$\{x, y, X_v, Y_v, H_x, H_y, H_{vx}, H_{vy}\} \qquad (2)$$

where (x, y) represent the center location of the head ellipse, (X_v, Y_v) represent the motion velocity, (H_x, H_y) are the lengths of the half axes, and (H_{vx}, H_{vy}) are the corresponding scale changes on the axes.

The key idea of a particle filter is to use a set of properly weighted random samples, $\{s_t^{(n)}, \eta_t^{(n)} \mid n = 1 \ldots N\}$, to approximate the posterior density function $P(S_t|Z_t)$, where $s_t^{(n)}$ is a sample of the state vector, $\eta_t^{(n)}$ the corresponding weight, and Z_t the observation vector.

0-7803-9332-5/05/$20.00 ©2005 IEEE

The proposed tracker performs the following three steps in tracking the head: *Sample Selection*, *Prediction* and *Updating*.

Sample Selection. Based on the weight associated with each sample (particle), N particles are drawn from the set $\{s_{t-1}^{(i)}\}_{i=1,2,3...N}$ at time t-1 with replacement to produce the sample set at time t. The larger the weight, the more likely a sample will be chosen to appear in the new sample set.

Prediction. Each sample is propagated according to the following system model:

$$S_t = AS_{t-1} + N_{t-1} \tag{3}$$

N_{t-1} is a random vector drawn from the noise distribution of the system. In this work, we use a simple system model, where the head motion is assumed to be constant velocity in translation and scaling, as in [4].

Update. Two new weights will be computed separately for each sample based on the new observations: one is based on the intensity gradient of the sample region; another is based on the color histogram difference between the measured color distribution and the model color distribution (computed at the previous time instant). Then the mean state, which specifies the tracked head, is estimated by

$$E[S] = \sum_{n=1}^{N} \eta^{(n)} s^{(n)}$$

In which $\eta^{(n)}$ is the final weight obtained by averaging the gradient weight and color histogram weight by

$$\eta^{(n)} = \alpha\eta_g^{(n)} + (1-\alpha)\eta_c^{(n)} \tag{1}$$

Currently we simply average the gradient weight and color histogram weight, so α is 0.5.

2.2. Gradient Estimation Based on the Head Shape

If we have obtained the contour of the head, then computing the gradient along the head contour provides a good measure for distinguishing the head from the background. Assuming that the ellipse in the model matches the head contour, then, for a particular state sample specified by s, the normalized sum of the gradient magnitude around the ellipse boundary is computed as:

$$\psi_g(s) = \frac{1}{Ns} \sum_{i=1}^{N_{(H_x, H_y)}} g(x_i, y_i)$$

where $g(x_i,y_i)$ is the intensity gradient of pixel (x_i,y_i) located at the boundary of ellipse specified by s. And N_s is the number of pixels on the perimeter of the ellipse.

Since an ellipse does not accurately describe the contour of the head, to make the above gradient estimate more useful in case of the inaccurate modeling, the gradient at pixel (x_i,y_i) is established as the maximum gradient by a local search along the normal direction:

$$g(x_i, y_i) = \max_{(x_n, y_n)\in L_n} \{g(x_n, y_n)\}$$

where L_n represents the normal line, (x_n, y_n) is the coordinate of the points that are located on the normal line. (x_n, y_n)must satisfy the following criterion :

$$\sqrt{(x_n - x_i)^2 + (y_n - y_i)^2} < SeachRange$$

$$y_n = \frac{(y_i - C_y) * H_x^2}{(x_i - C_x) * H_y^2} * (x_n - x_i) + y_i$$

The first formula specifies that the distance in the normal direction between point (x_n, y_n) and (x_i, y_i) must be within a certain search range. This will help to restrict our search around the head contour and avoid hitting other distracting points on the background (or within the face) which have big intensity gradients. The second one is the normal line equation at point (x_i, y_i). (C_x, C_y) denotes the ellipse center.

A simple operator is used to compute the gradient in x-direction and y-direction for pixel (x_n,y_n):

$$g_x(x_n,y_n) = I(x_n-2,y_n) + 2*I(x_n-1,y_n) - 2*I(x_n+1,y_n) - I(x_n+2,y_n)$$

$$g_y(x_n,y_n) = I(x_n,y_n-2) + 2*I(x_n,y_n-1) - 2*I(x_n,y_n+1) - I(x_n,y_n+2)$$

And finally the gradient at point (x_n, y_n) is computed as

$$g(x_n, y_n) = \sqrt{g_x^{\,2}(x_n, y_n) + g_y^{\,2}(x_n, y_n)}$$

The above local search provides a potentially significant improvement over the simple gradient method in [5]. This step is especially helpful in the particle filter setting, since some of state samples may deviate from the face even if the head is a perfect ellipse.

2.3. Color Distribution Model

Following [4], we use a histogram-based color model. The color distribution is represented by a color histogram calculated in the RGB space using 8*8*8 bins. Pixels which are closer to the region center are given higher weights specified by:

$$k(r) = \begin{Bmatrix} 1 - r^2 : r < 1 \\ 0 : otherwise \end{Bmatrix}$$

This helps to put more weight on the center pixels than on the boundary pixels, since the boundary pixels may not correspond so well to the actual head.

The color distribution $p_y = \{p_y^{(u)}\}_{u=1,2...m}$ of a region R at location y is calculated as (see [4] for details):

$$p_y^{(u)} = f \sum_{X_i \in R} k\left(\frac{\| y - x_i \|}{a}\right) \delta\left[h(x_i) - u\right]$$

where δ is the Kronecker delta function and $h(x_i)$ measures the color of the pixel at location x_i . To measure the similarity between the newly observed image and the target model, Bhattacharyya coefficient is computed between these two distributions:

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

p stands for the color histogram of a sample hypothesis in the newly observed image, and q represents the color histogram of the target model. The larger ρ is, the more similar the distributions are.

2.4. Determine the Sample Weights

Under the assumption that the true contour of the face would typically generate a larger gradient value than a contour that is randomly overlaid on the face, we employ the following simple method to relate the computed gradient to the sample weight

$$G_g^{(n)} = \frac{1}{\sqrt{2\pi}\,\sigma_g} e^{-\frac{(1/g)^2}{2\sigma_g^2}} \tag{13}$$

where σ_g^2 is the variance controlling the model, and g is calculated in Sect. 2.2. This Gaussian distribution matches to the intuitive requirement that a larger g should produce a larger weight. Although other model can achieve similar objective, the Gaussian model is found to be effective in this work.

For the color histogram model, we first calculate color histogram inside the entire sample ellipse, and then the Bhattacharyya coefficient is computed, which is then again fed into a Gaussian distribution to obtain color weight as follows:

$$G_g^{(n)} = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{d^2}{2\sigma_c^2}} = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{(1-\rho[p_{s(n)},q])}{2\sigma_c^2}}$$

The above two weights are averaged to obtain a final weight as in Eq. (1).

3. EXPERIMENTS AND EVALUATIONS

To evaluate the performance of the proposed tracker, we test with both synthetic and real sequences. We evaluate the method by invoking either or both of the cues in different experiments. That is, the tracker has three modes of operations: one based on the gradient model alone, one based on the color model alone (effectively the method of [4]), and one based on both models (the proposed method). The experiments are performed on a 2.8GHz PC under Windows XP, the 24-bit RGB sequences are of 320*240 resolution.

3.1. Experiments on Synthetic Sequences

A synthetic sequence is created by overlaying a moving head (62*41 pixels) with constant velocity in translation on a static image background. To show the tracking results, for each frame, a deformed contour is drawn around the ellipse boundary specified by the tracked mean state. The deformed contour (plotted in green) consists of pixels with maximum gradient along normal direction of the mean state ellipse.

Figures 1 and 2 illustrate that, when the background is relatively uniform, the gradient model alone performs as good as the other two; but when the environment is cluttered and has complex structures, the gradient tracker is distracted a little bit by the clutters in the background. In this case, the combined model with both color and gradient information gives the best performance. (For the given synthetic sequence, since the background is almost colorless, using only the color model can also achieve good tracking results, which are hence not shown here).

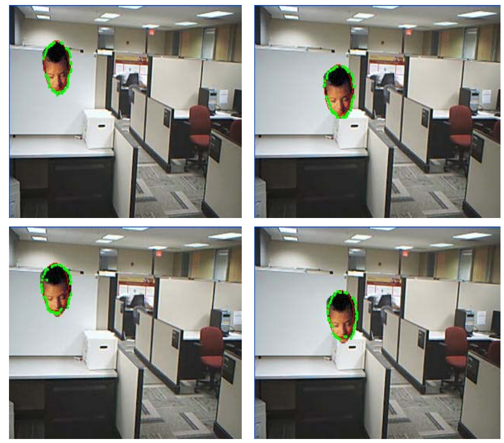

Figure 1. Tracker performance under different modes of operations, on the synthetic sequence. Top row: with the gradient model alone. Bottom row: with both the color and the gradient models. In this example, since the background is relatively uniform, there is no big difference between the two modes of operations.

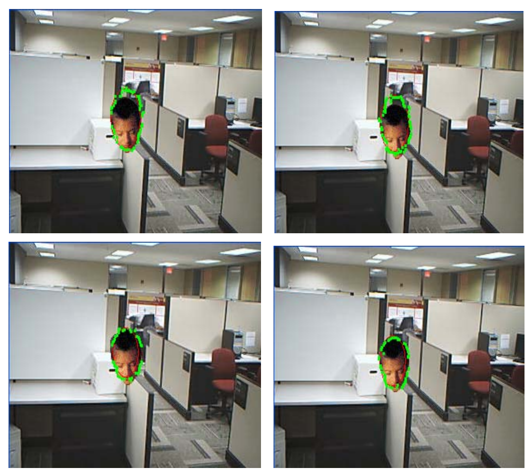

Figure 2. Tracker performance under relatively cluttered background. Top row: with the gradient model alone. Bottom row: with both the color and the gradient models. In this example, employing the color model helps greatly to achieve more accurate tracking.

To further examine the tracker's accuracy quantitatively, we compare the tracked data (i.e., the ellipse center (x, y) and size (H_x, H_y)) from the three modes (color only, gradient only, color and gradient) against the corresponding ground truth. Sample results are shown in Fig. 3, where the result from the color model alone is omitted since it is similar to the result from the combined model, for reasons explained earlier. We found that, due to the cluttered background, the gradient model alone exhibits a larger tracking error than the combined model. (Note that, in the synthetic data, there is no scale change, so ideally the tracked H_y should be a constant. In Fig. 3, the better performance of the combined model manifests as the relatively smaller variations of the H_y values.)

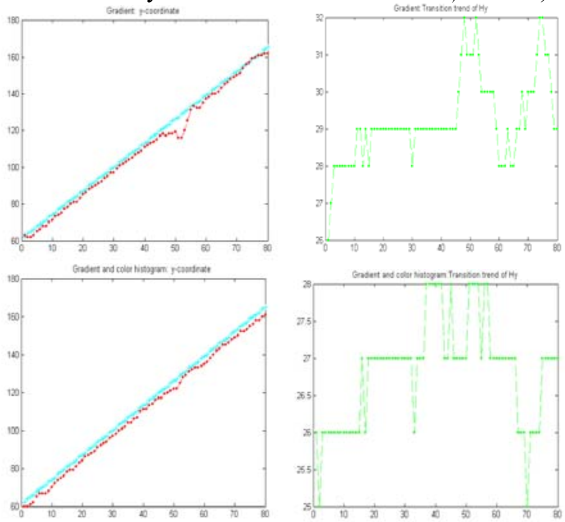

Figure 3. Quantitative comparison between the gradient model and the combined model. Left column: tracked y-coordinates (dashed and red) and the ground truth (solid and cyan). Right column: tracked H_y values. Top row: from the gradient model alone. Bottom row: from the combined model.

3.2. Experiments on real sequences

To demonstrate the tracker's robustness, we conducted extensive experiments on real image sequences. Figure 4 shows a sample result, with comparison with the color histogram tracker of [4]. The result from using only the gradient model is also given, as comparison. It is obvious that the proposed method handles much better the challenging situations including full 360-degree rotation and camera zooming. This and other experimental results show that, the color model contributes to the tracker by ignoring cluttered objects background, while the gradient model help to compensate for the dramatic color change due to head rotation. Thus the combined model produces on average the best results.

4. CONCLUSION AND FUTURE WORK

We proposed a head tracker using both the color and the gradient information in a particle-filter-based framework. Experiments show that these two complementary cues contribute to a more robust tracker, which handles difficult situations that a tracker based on either of the cues alone cannot handle well. Future work along this direction includes a complete investigation of the tracker's performance under multiple human subjects with occlusion.

REFERENCES

[1] C.Shen, A.van den Hengel, and A. Dick, "Probabilistic Multiple Cue Integration for Particle Filter Based Tracking", *Proc. VIIth Digital Image Computing: Techniques and Applications*, Sydney, Australia Dec.2003.

[2] A. Colmenarez, R. Lopez and T. Huang, "3D Model-based Head Tracking", *Visual Communication and Image Processing*, San Jose, CA, 1997.

[3] Z.Zivkovic,F.van der Heijden, "A stabilized adaptive appearance changes model for 3D head tracking", *Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, Vancouver, Canada, July 2001.

[4] K.Nummiaro, E. Koller-Meier, and L.V.Gool, "Object Tracking with an Adaptive Color-Based Particle Filter", *Proc. Symposium for Pattern Recognition of the DAGM*, pp. 591-599, 2003

[5]S. Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histogram", *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pp. 232-237 ,June 1998.

[6] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", IEEE International Conference on Image Processing (ICIP 2002), Rochester, USA, September, 2002.

[7] J. Yang and A. Waibel, "A Real-time Face Tracker", Proc. of WACV 1996, pp. 142-147.

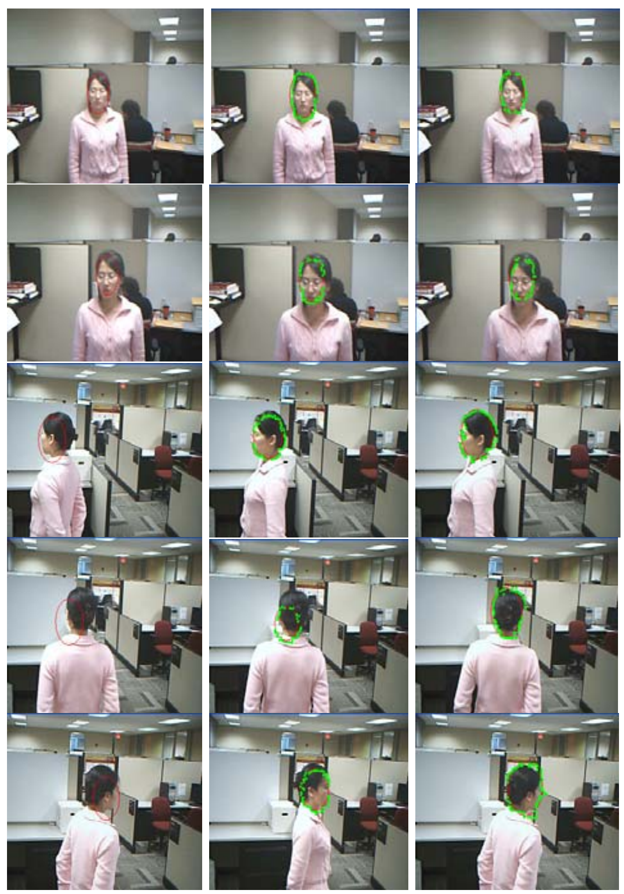

Figure 4. Sample results from a real sequence. Left column: Color histogram tracker of [4]. Center column: the proposed method with only the gradient component. Right column: the proposed method with both color and gradient components. Note that, the first two frames is an example of camera zooming, while the other frames depict full 360-degree rotation.