A Robust Real-Time Face Tracking Algorithm

Richard J. Qian M. Ibrahim Sezan Kristine E. Matthews

Sharp Laboratories of America
5750 N.W. Pacific Rim Blvd.
Camas, WA 98607
Email: (qian, sezan, matthews)@sharplabs.com

Abstract

This paper presents a statistics-based method for estimating the position and size of a face in complex background. Face position and size are estimated based on robust statistical measurements which are derived from two one-dimensional histograms obtained by projecting the result of skin color filtering. The proposed algorithm also utilizes a linear Kalman filter and a simple nonlinear filter to perform smooth tracking and remove jitter. The algorithm has been implemented and tested under a wide range of real-world conditions. It has consistently provided performance which satisfies the following requirements: 1) able to automatically determine the initial position and size of a face and track it in complex background; 2) insensitive to partial occlusions and shadows; 3) insensitive to face orientation and scale changes; and 4) insensitive to lighting condition changes. In addition, the algorithm is computationally simple so that it can be executed in real-time.

1. Introduction

Robust real-time face tracking has many applications in advanced video communication and human computer interaction. In a mobile video phone application, for example, face tracking can be used to reduce communication bandwidth by locating and transmitting only the fraction of a video frame which contains the speaker's face. In a human computer interaction application, face tracking can be used to direct the computer's attention to a user and increase the probability of the computer to correctly recognize the user's facial expressions, gestures, or speech.

Face tracking has been studied by many researchers in the past. Here we only refer to some of the recent work. Eleftheriadis and Jacquin proposed a face tracking algorithm in [2] which is based on the frame difference due to a moving head and the assumption that the outline of a person's head is elliptical. In a mobile video phone application, however, there usually exists substantial movement of the background. Therefore the head region may be cluttered or obscured by the other moving regions. The elliptical outline assumption may also break when partial occlusions or shadows occur or a person wears a hat. In addition, the proposed algorithm requires intensive computation making it difficult to use in real-time applications. Hager and Belhumeur proposed a general-purpose tracking algorithm in [4] which modifies the correlation-based algorithms by explicitly modeling different illumination conditions via basis images. One limitation of the algorithm is that it requires the initial position of a face as input to perform face tracking. Also since the basis images are constructed from unoccluded front-view faces, the method may be sensitive to partial occlusions and the change of face orientation. McKenna and Gong proposed a face tracking algorithm in [5] which is based on moving region detection and front-view face detection. Again the performance of moving region detection may be hindered by background motion in a mobile camera situation. Also the face detection algorithm detects only front-view faces and may be sensitive to partial occlusions. With Raja, the authors proposed another tracking algorithm in [6] which is based on color. Color region grouping is suggested to locate and track objects but no detail is given on how robust the grouping process is in practice. Yang and Waibel proposed a face tracking algorithm in [8] which is based on skin color filtering. Unfortunately, the authors failed to present any concrete procedure for locating a face based on the results of skin color filtering. Moreover, since their method employs a simple tracking mechanism, a tracked face may jitter in the presence of image noise and lighting fluctuation.

What is desired, therefore, is a face tracking algorithm which is 1) able to automatically determine the initial location and size of a face and track it in complex background; 2) insensitive to partial occlusions and shadows; 3) insensitive to face orientation and scale changes; and 4) insensitive to lighting condition changes. In addition, the algorithm should be computationally

0-8186-8821-1/98 $10.00

simple so that it can be executed in real-time. In this paper, we present a face tracking algorithm to accomplish the above requirements. The proposed algorithm has also been implemented and tested under a wide range of real-world conditions.

In Section 2, we first present an overview of our face tracking algorithm. We then describe each individual step. Emphasis will be given to the procedures which locate and track a face using the results of skin color filtering. In Section 3, we describe our experiments and present an example of our results. We also describe the memory requirement and computational cost of our implementation. Finally in Section 4, we discuss certain aspects of our algorithm and point out some future work.

2. Color-Based Face Tracking

Our algorithm locates and tracks a face in a color video sequence based on skin color information. Given an input frame, the algorithm first extracts skin pixels by color filtering. Based on the result from color filtering, it constructs two one-dimensional (1D) histograms which represent the distributions of the skin pixels in two orthogonal directions, respectively. The instantaneous center position and size of the tracked face are estimated based on statistical measurements derived from the histograms. The estimated values of the face center position and size are then fed into a tracking mechanism which consists of a linear Kalman filter and a simple nonlinear filter. Finally the output values from the tracking module are used as the tracked center position and size of the face. Figure 1 shows a block diagram of our algorithm.

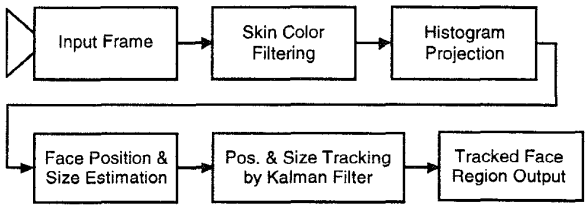

Figure 1. Block diagram of the proposed real-time face tracking algorithm.

2.1. Skin color filtering

Researchers have discovered that human skin colors even from different races tend to cluster in a compact region in certain transformed two-dimensional (2D) color spaces, see e.g., [3, 8, 1]. Such transformed 2D color spaces include 1) (R_g, R_y) in the log-opponent color space; 2) the chromaticity space (r, g); and 3) (U, V) in the YUV color space. The purpose of employing the transformed color spaces is to reduce the dependency on intensity or brightness, so that skin colors of different darkness or under different lighting intensity are drawn close to each other after they are mapped into the transformed color spaces. In general, the log-opponent transformation and the chromaticity transformation achieve better reduction of intensity dependency than the YUV transformation. In our algorithm, we use the chromaticity transformation because it is computationally simpler than the log-opponent transformation. For a given pixel (R, G, B), the chromaticity transformation is defined as: $r = R / (R+G+B)$ and $g = G / (R+G+B)$.

Gaussian distributions may be used to model the skin color distribution [5, 4]. For simplicity, we currently model the cluster of skin colors in the chromaticity space as a single 2D Gaussian distribution $p(\mu_r, \mu_g, \sigma_r, \sigma_g)$. The parameters ($\mu_r$, μ_g, σ_r, σ_g) may be obtained by automatic training or interactive calibration based on skin color samples. In general, we found that for a given camera the skin color model needs to be established only once and it remains valid under a wide range of conditions.

The skin color filtering process maps an input color image into a color filtering map in which most of the skin pixels are marked toward 1 and most of the non-skin pixels are marked toward 0. For a given pixel (R, G, B), the skin color filter is defined as follows:

$$f_{CF}(R,G,B) = \exp\left[-\frac{(r-\mu_r)^2}{2\sigma_r^2} - \frac{(g-\mu_g)^2}{2\sigma_g^2}\right],$$

where (r, g) are the chromatic colors of the pixel as defined earlier. To reduce noise effect, the color filtering result may be further processed by a morphological filter.

The default skin color model may be automatically adjusted and therefore adapted to a particular user and lighting condition during an actual tracking session. One way to perform the color adaptation is first to collect pixel samples whose filtering scores based on the default skin color model exceed a prescribed threshold. Then the system updates the mean and standard deviation of the skin color model based on the collected actual skin pixel samples. Such an adaptation procedure may be invoked continuously or periodically during the entire course of tracking.

2.2. A Statistics-based method for estimating face position and size

Based on the result from skin color filtering, our algorithm constructs two 1D histograms by projecting the filtering map along its x and y direction, respectively. In other words, the color filtering scores of all the pixels in the input frame are summed up along the x and y direction to

form two separate 1D histograms. The histograms, therefore, represent the spatial distributions of the skin pixels along the corresponding axes. Figure 2 illustrates an ideal skin color filtering map where there is only one face in the input image, and the corresponding projection histograms.

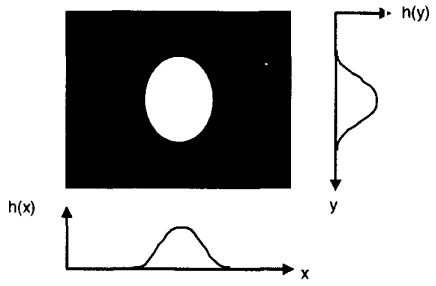

Figure 2. Two 1D histograms constructed by projecting the skin color filtering map along the x and y direction, respectively.

The instantaneous center position and size of a face in the image can be estimated based on statistical measurements derived from the two 1D projection histograms. For example, a simple method estimates the center position and size of a dominant face in an input image using the sample means and standard deviations of the distributions. More specifically, let $h_x(i)$, $i = 0, 1, ...,$ and $h_y(i)$, $i = 0, 1, ...,$ denote the elements in the projection histograms along the x and y direction, respectively. Then the face center position (x_c, y_c) and face width and height (w, h) may be estimated as:

$$x_c = \frac{\sum_i x_i h_x(i)}{\sum_i h_x(i)}, \quad y_c = \frac{\sum_i y_i h_y(i)}{\sum_i h_y(i)},$$

$$w = \alpha \left[\frac{\sum_i (x_i - \mu_x)^2 h_x(i)}{\sum_i h_x(i)} \right]^{1/2}, \quad h = \beta \left[\frac{\sum_i (y_i - \mu_y)^2 h_y(i)}{\sum_i h_y(i)} \right]^{1/2},$$

where α and β are constant scaling factors.

However, the face center position and size derived from the sample means and standard deviations may be biased in the cases where other faces and/or skin-color-like objects appear in the scene. It is therefore necessary to develop a more robust procedure to address this problem. In this paper, we propose to use robust statistical estimation routines to achieve robust measurements for face center position and size [7]. More specifically, the center position of a dominant face in an input image is estimated based on the robust (trimmed) means of the two 1D projection histograms in the x and y directions. The routine for estimating the center position in either x or y direction is as follows. Figure 3 illustrates such a process.

Step 1. Compute sample mean μ and sample standard deviation σ based on all the samples of the distribution.

Step 2. Let $\mu_t(0) = \mu$ and $\delta = max(a * \sigma, b * sample\text{-}space\text{-}width)$ where a and b are scaling factors, e.g., $a = 1.0$ and $b = 0.2$, and sample-space-width is the image-width and image-height in the x and y direction, respectively.

Step 3. Compute trimmed mean $\mu_t(k+1)$ based on the samples within the interval $[\mu_t(k) - \delta, \mu_t(k) + \delta]$.

Step 4. Repeat Step 3 until $| \mu_t(k+1) - \mu_t(k) | < \varepsilon$ where ε is tolerance, e.g., $\varepsilon = 1.0$. Denote the converged mean as μ^*.

Step 5. Let center-position = μ^*.

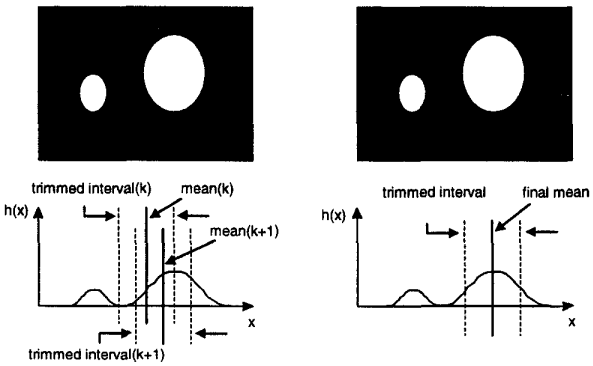

Figure 3. Robust mean estimation for locating the center position of a dominant face.

In addition to the robust estimation of face center position, we propose the following routine for robust estimation of face size. The method first re-projects the color filtering result in a neighborhood of the located center. It then derives the face size based on the robust(trimmed) standard deviation. Given the robust mean μ^* and δ obtained from the above robust estimation routine for center, the routine for estimation the size in either x or y direction is as follows.

Step 1. Construct a clipped projection histogram $\mathbf{H}^{clip}$ by projecting the color filtering map within the range $[\mu^*_{opp} - \Delta, \mu^*_{opp} + \Delta]$ in the opposite direction, where μ^*_{opp} is the robust mean in the opposite direction and Δ determines the number of samples used in the calculation.

Step 2. Based on $\mathbf{H}^{clip}$, *compute the trimmed standard deviation* σ_t *based on the samples within the interval* $[\mu^* - \delta, \mu^* + \delta]$.

Step 3. If $h^{clip}(\mu^* + d * \sigma_t) \geq g * h^{clip}(\mu^*)$ *or* $h(\mu^* - d * \sigma_t) \geq g * h^{clip}(\mu^*)$ *where, e.g.,* $d = 1.0$ *and* $g = 0.4$, *then increase* σ_t *until the condition is no longer true.*

Step 4. Let size $= c * \sigma_t$ *where c is a scaling factor, e.g.,* $c = 2.0$.

2.3. Tracking face position and size using linear Kalman filter and nonlinear filter

In our face tracking algorithm, the estimated instantaneous face center position (x_c, y_c) and face size (w, h) are fed into a linear Kalman filter followed by a simple nonlinear filter. The Kalman filter is used to smooth the temporal trajectories of the face center position (x_c, y_c) and size (w, h). It can also be used to predict the change of those quantities. The nonlinear filter is used to completely remove jitter due to image noise and lighting fluctuation. We model the temporal changes of face position (x_c, y_c) and size (w, h) as four independent piecewise-constant 2D translations within the image plane. Let $\mathbf{x}(t)$ denote the true four-dimensional velocity vector at time t, which is to be estimated. The system model for tracking is:

$$\mathbf{x}(t+1) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t),$$
$$\mathbf{z}(t+1) = \mathbf{H}(t+1)\mathbf{x}(t+1) + \mathbf{v}(t+1),$$

where $\mathbf{z}(t)$ is the observed instantaneous velocity vector, $\mathbf{w}(t)$, $\mathbf{v}(t)$ are white noise, and $\mathbf{F}(t) \equiv \mathbf{I}$, $\mathbf{H}(t) \equiv \mathbf{I}$ for piece-wise constant motion. The Kalman predictor is:

$$\hat{\mathbf{x}}(t+1|t) = \mathbf{F}(t)\hat{\mathbf{x}}(t|t), \quad \hat{\mathbf{x}}(0|0) = 0,$$
$$\hat{\mathbf{z}}(t+1|t) = \mathbf{H}(t+1)\hat{\mathbf{x}}(t+1|t).$$

The Kalman corrector is:

$$\hat{\mathbf{x}}(t+1|t+1) = \hat{\mathbf{x}}(t+1|t) + \mathbf{K}(t+1)\Delta\mathbf{z}(t+1|t),$$
$$\Delta\mathbf{z}(t+1|t) = \mathbf{z}(t+1) - \hat{\mathbf{z}}(t+1|t),$$

where $\mathbf{K}(t+1)$ is the Kalman gain. The Kalman gain is computed as:

$$\mathbf{K}(t+1) = \mathbf{P}(t+1|t)\mathbf{H}^{\mathrm{T}}(t+1)[\mathbf{H}(t+1)\mathbf{P}(t+1|t)\mathbf{H}^{\mathrm{T}}(t+1) + \mathbf{R}(t+1)]^{-1}.$$

The covariances are updated as:

$$\mathbf{P}(t+1|t) = \mathbf{F}(t)\mathbf{P}(t|t)\mathbf{F}^{\mathrm{T}}(t) + \mathbf{Q}(t), \quad \mathbf{P}(0|0) = \mathbf{P}_0,$$
$$\mathbf{P}(t+1|t+1) = [\mathbf{I} - \mathbf{K}(t+1)\mathbf{H}(t+1)]\mathbf{P}(t+1|t),$$

where $\mathbf{Q}(t) = E[\mathbf{w}(t)\mathbf{w}^{\mathrm{T}}(t)]$, $\mathbf{R}(t) = E[\mathbf{v}(t)\mathbf{v}^{\mathrm{T}}(t)]$ and $\mathbf{P}_0 = E[\mathbf{x}(0)\mathbf{x}^{\mathrm{T}}(0)]$.

To completely remove jitter, the output of the Kalman filter is further fed into a simple nonlinear filter. The nonlinear filter zeros out any position and size changes of the tracked face which are smaller in magnitude than some prescribed thresholds. More specifically, for the estimated change of each of the four tracked quantities (x_c, y_c) and (w, h), the jitter removing filter is defined as:

$$f_{JR}(\hat{x}) = \begin{cases} \hat{x} & \text{if } \hat{x} > \lambda \\ 0 & \text{else} \end{cases}$$

where λ is a prescribed threshold.

3. Experiments

We have implemented the proposed face tracking algorithm on Sun workstations and on PCs. The resulting face tracking systems have then been tested in different locations with various backgrounds and lighting conditions on a large number of people (150+) with a wide range of skin tones. The systems have consistently provided performance which satisfies the requirements listed in the beginning of this paper. Figure 4 shows an example of the results in each step employed in our face tracking algorithm.

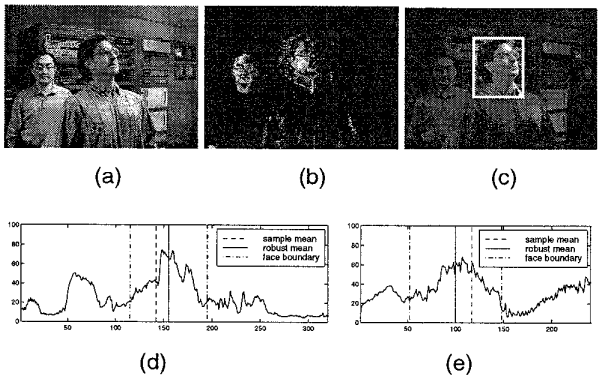

Figure 4. An example of the results in each step of the proposed face tracking algorithm. (a) Input image. (b) Skin color filtering result. (c) Tracked dominant face. (d) Projection histogram in the x direction and the estimated face center (robust mean) and face boundary. (e) Projection histogram in the y direction and the estimated face center (robust mean) and face boundary.

The proposed face tracking algorithm requires a very small amount of memory and a relatively small amount of computation. Its essential memory consumption is under 2K and 3K bytes for frame resolutions of 160×120 pixels and 320×240 pixels, respectively. Its computational cost is (34,000 additions + 20,000 multiplications) and (135,000 additions + 78,000 multiplications) per frame for frame resolutions of 160×120 pixels and 320×240 pixels, respectively. The actual frame rate of our system on a 168MHz Sun workstation is 30 and 20 frames per second for frame resolutions of 160×120 pixels and 320×240 pixels, respectively. The above frame rates include the screen display time for both input and output video. On a 200MHz PC, the frame rate is largely limited by the screen display time. In fact, we could not detect any change in frame rate when we ported our face tracking module into the PC system.

4. Discussion

This paper presents a real-time face tracking algorithm which is based on robust statistical measurements derived from the result of skin color filtering. One unique feature of the proposed algorithm is that it constructs two 1D histograms by projecting the color filtering results in two orthogonal directions. The projection histograms represent the spatial distributions of the skin pixels along the corresponding directions. The advantage of constructing such projection histograms is that it reduces the computational complexity of locating a face from a 2D problem to two 1D problems. It further allows robust statistical estimation routines to be applied to the two 1D distributions. In the proposed algorithm, the color filtering map is chosen to contain continuos values which represent the probabilities of belonging to skin for the corresponding pixels in the input image. Alternatively, the color filtering map may be chosen to contain binary values which explicitly label all the pixels into either skin pixels or non-skin pixels via hard thresholding. But using a hard threshold may cause some pixels to switch their classifications between frames due to their marginal filtering values. Such abrupt switching may lead to considerable changes in the filtering result and the corresponding projection histograms, which may then introduce jitter. Therefore the advantage of adopting a continuos filtering map is that the corresponding projection histograms may be smoother and remain stable between frames. The proposed algorithm also utilizes a linear Kalman filter and a simple nonlinear filter to perform smooth tracking and remove jitter. In addition to automatically determining the initial face location and size, our algorithm consistently tracks the face in scenes with complex backgrounds. The algorithm is insensitive to a variety of factors including partial occlusions, shadows, face orientation, and changes in scale and lighting conditions, therefore offering robust performance in a wide range of real-world scenarios and applications. The utility of our algorithm is further enhanced by its computational simplicity which allows real-time execution.

There are a number of directions in which future work may be continued. One direction is to develop a unified method which incorporate multiple visual cue processes such as color filtering, motion detection and facial pattern detection. Adding motion and facial pattern cues to a color-based method may help to resolve the ambiguity between faces and skin-color-like background and/or objects. Another direction is to improve the robustness for estimating the size of a face. Our current size estimation routine provides correct face size information as long as noise is small in the color filtering result. But when there is significant amount of noise, the routine may generate inaccurate size information. Yet another direction for future work is to develop more sophisticated skin color models and color adaptation procedure to cope with dramatic lighting condition changes such as from indoor to outdoor or vice versa.

References

[1] D. Chai and K. N. Ngan, "Locating facial region of a head-and-shoulders color image," *Proc. International Conference on Face and Gesture Recognition*, pp. 124-129, 1998.

[2] A. Eleftheriadis and A. Jacquin, "Automatic face location detection and tracking for model-assisted coding of video teleconferencing sequences at low bit-rates," *Signal Processing: Image Communication*, No. 7, pp. 231-248, 1995.

[3] M. M. Fleck, D. A. Forsyth and C. Dregler, "Finding naked people," *Proc. European Conference on Computer Vision*, pp. 593-602, 1996.

[4] G. D. Hager and P. N. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination," *Proc. Computer Vision and Pattern Recognition*, pp. 403-410, 1996.

[5] S. J. McKenna and S. Gong, "Tracking faces," *Proc. International Conference on Face and Gesture Recognition*, pp. 271-276, 1996.

[6] Y. Raja, S. J. McKenna and S. Gong, "Tracking and segmenting people in varying lighting conditions using color," *Proc. International Conference on Face and Gesture Recognition*, pp. 228-233, 1998.

[7] R. R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing*. Statistical Modeling and Decision Science Series. Academic Press, 1997.

[8] J. Yang and A. Waibel, "Tracking human faces in real-time," *Proc. IEEE Workshop on Applications of Computer Vision*, 1996.