POSTER: Augmentation of Mean-Shift Method to Facilitate Face Tracking

Albert Akhriev
IBM Dublin Software Lab
Building 6, IBM Technology Campus, Damastown Industrial Park, Mulhuddart, Dublin 15, Ireland
aaahaaah@hotmail.com

ABSTRACT

In this paper we introduce an improvement to the well known mean-shift color tracker. On each new frame we estimate the histogram resolution which provides the best separation between color distributions of the object and the background. The optimal resolution is derived from the principle of minimum uncertainty of foreground/background classification. The augmented mean-shift method with variable histogram resolution was applied to the task of face tracking.

Keywords

Mean-shift color tracker, histogram resolution, Renyi entropy.

1. INTRODUCTION

Creating interactive computer systems that are effective and easy to use is an important direction of modern research. For example, face and hand tracking software can be a part of a human-computer interaction system that creates the illusion of feedback between user and an OpenGL application running on a personal computer. In this paper we focus on the *face tracking* method based on the well known mean-shift color tracker [Com03].

Recent developments have shown the effectiveness of color based tracking algorithms for objects with variable appearance. These methods can be separated into geometric (contour, region), feature-based (color, texture) and hybrid ones. Feature-based and hybrid methods are of particular interest in recent years and can be further subdivided into distribution tracking [Com03, Goo02, McK99, Zha05] and feature classification [Ngu02, Bra98] methods.

Distribution tracking is a relatively robust process, which locks firmly on an object's sub-region over multiple frames; however, it might not follow the shape and orientation changes.

A typical feature classification method uses a model (e.g. histogram) to classify points of a new frame into object or non-object ones. Usually the model is deduced from past observations. The result of a classification is a 2D-array of weights or probabilities, where adjacent points with high values are aggregated together to form an object. This approach is geometrically flexible, but also error prone since it can produce false positives. If a distribution is dispersed across a feature space (e.g. histogram bins are not densely populated), a classification might be unreliable. Fortunately, face color distribution is usually compact.

We used the mean-shift tracker [Com03], which has been proved to be robust and computationally effective, as a reference. In order to reduce the sensitivity of color tracking methods to the presence of object-like features in the background, we came up with the following contributions:

1) We replaced the ad-hoc background utilization proposed in [Com03]. Instead, the histogram resolution, which provides the optimal separation between foreground and background, is computed in each new frame [Akh07].
2) Given estimation of object's location in a new frame, obtained by the mean-shift tracker, we generate a *probability map* (see Figure 2), where each pixel keeps the probability of object presence. Our aggregation method uses the probability map to make the final estimation of object's position.

2. COLOR TRACKING ALGORITHM

Optimal Histogram Resolution

Hereinafter we talk about color features only and color histograms as the distribution models. We assume that the reader is aware of the basic concept of the mean-shift tracking approach [Com03].

One possible way to find the optimal histogram resolution is to choose the one that minimizes statistical distance between foreground and background color distributions in the current frame, where object location is known. The best resolution found is used by the tracker in the next frame. The drawback of this approach occurs from unstable lighting conditions and low quality video output from cheap web-cameras. In contrast, our method takes into consideration a portion of the *next* frame, which likely contains the object being tracked.

Let us begin with some notation. At each moment t on the frame I_t the face occupies an *elliptic* region $\mathbf{r}_t$ obtained as the result of previous calculations. At the moment $t = 0$ initial region is given. Let us introduce the set of face or *foreground* points F_t and the set of *background* points B_t. These sets form two regions inside and outside of face respectively (left picture on Figure 1). The set $C_t = F_t + B_t$ includes all points of the *current* frame that participate in the tracking process. Let us define the set of points N_t of the *next* frame I_{t+1} covered by the set C_t shifted on the vector of mean interframe displacement $T_t = (dx,dy)$ (right picture on Figure 1). We do not know the exact face position at the moment t+1, but we assume that it lies somewhere inside N_t. A color vector $\mathbf{c} = (R,G,B)$ is given at each image point. Color histograms H_f, H_b and H_n are constructed on the point sets F_t, B_t and N_t respectively. Let us denote a histogram entry for a color $\mathbf{c}$ as H($\mathbf{c}$), and the number of samples as |H|.

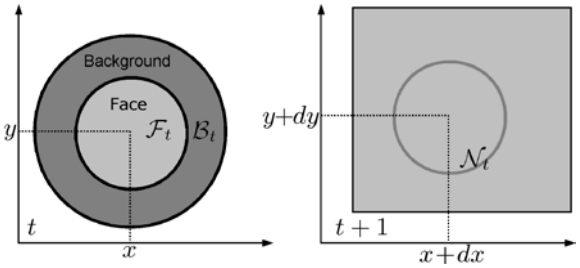

Figure 1. Regions on the current frame (left) and candidate region on the next frame (right).

The Idea of Optimal Histogram Synthesis

A color component R, G or B varies within the interval [0,255]. We define *histogram resolution* as the number of bins this interval is divided into. Optimal resolutions of channels R_{bin}, G_{bin} and B_{bin} are obtained by minimization of classification uncertainty

$$U = \min_{R_{bin},G_{bin},B_{bin}} \sum_{\mathbf{c}\in N_t} H_n(\mathbf{c})\, E\Big(H_b(\mathbf{c}), H_f(\mathbf{c})\Big), \quad (1)$$

where $H_n(\mathbf{c})$ is the number of points of the set N_t with color $\mathbf{c}$, and $E(,)$ is the entropy (uncertainty measure) of classification of a point with color $\mathbf{c}$. The entropy function receives the numbers of samples with color $\mathbf{c}$ in the background and foreground distributions as its arguments. Summing in (1) is carried out over all non-zero entries of the histogram H_n.

The idea of (1) is to minimize classification uncertainty over a region of the next frame, which contains an object being tracked, using foreground and background histograms accumulated so far at the current frame. We assume that histograms of the best resolution separate foreground and background points on the *next* frame with the least uncertainty. This is not necessarily true, but the heuristic works well in practice.

In order to reduce the mathematical complexity, we choose equally populated sets B_t and F_t ($|H_b| = |H_f|$) by proper selection of the background ellipse (the outer ellipse on Figure 1) or downscale the largest histogram.

Minimization of (1) is achieved by direct enumeration of histogram resolutions R_{bin}, G_{bin} and B_{bin}. We divide the interval [0,255] into 4, 8, 16 and 32 bins. Three channels give 4^3=64 partitions of color space. The best partition minimizes (1).

Entropy

In this paper, *point classification* means assignment of a background p_b and a foreground p_f probabilities to a point with color $\mathbf{c}$ on the next frame. Renyi entropy of such classification is given as follows:

$$E(\mathbf{c}) = \ln\Big(p_b^2(\mathbf{c}) \quad p_f^2(\mathbf{c})\Big), \quad (2)$$

where empirical probabilities can be defined, for a while, via frequency of occurrence in histograms H_b and H_f respectively, e.g. $p_f = H_f(\mathbf{c})/(H_b(\mathbf{c})+H_f(\mathbf{c}))$. Both Shannon and Renyi entropies produce similar results, but there exists efficient approximation to Renyi entropy: $E = const\ (p_b\, p_f + 2.15544\ (p_b\, p_f)^2)$.

In practice formula (2) does not work correctly because many histogram bins are populated with only few samples. In [Akh07] we have shown that more realistic results can be obtained by averaging the entropy over an unknown "true" state. The important consequence is that the average entropy significantly increases, comparing with the theoretical one (2), as the number of samples in a histogram bin becomes too small. *The optimal histogram resolution compromises between good separability of background and foreground distributions (high resolution) and low classification uncertainty (low resolution, many samples in a bin).*

The first term in Renyi entropy approximation gives the following expression after averaging (see [Akh07])

$$\bar{E}(\mathbf{c}) \approx \frac{const \cdot (1+H_b(\mathbf{c}))(1+H_f(\mathbf{c}))}{(2+H_b(\mathbf{c})+H_f(\mathbf{c}))(3+H_b(\mathbf{c})+H_f(\mathbf{c}))}.$$

We call it the *discrete entropy*. Discrete entropy participates in minimization (1). Then the optimal histogram resolution is used to run the mean-shift tracker. Similarly we calculate expectation of empirical probability, which takes into account a (potentially) small number of samples in a histogram bin, [Akh07]

$$\bar{p}_f(\mathbf{c}) = \left(1 + H_f(\mathbf{c})\right) / \left(2 + H_b(\mathbf{c}) + H_f(\mathbf{c})\right). \quad (3)$$

Blob Aggregation

The original mean-shift tracker [Com03] does not support plane rotation of an object. On the other hand, the cam-shift method [Bra98] provides flexible adaptation of an elliptic face, but often fails when the subtle balance between optimal window size and actual space distribution of skin-like points is broken. We have found that ideas from both methods can be unified into a better approach:

1. Given parameters of a face ellipse on the current frame I_t (center position, large and small semi-axes, angle between x axis and large semi-axis), find the optimal histogram resolution.
2. Compute foreground (H_f) and background (H_b) histograms with optimal resolution. Feed the histogram H_f into the mean-shift tracker.
3. Run the mean-shift tracker 9 times with scale factors {0.9, 1.0, 1.1} in the directions of ellipse axes and pick up the best estimation [Com03].
4. Keeping the same value of ellipse area, change estimated ellipse so that the ratio between the large and the small semi-axes does not exceed 1.2 [Bir98, Bra98]. The latter improves the overall stability of the color face tracker.
5. Take in turn all points of the set N_t and calculate foreground probability (3) at each point. Points outside N_t receive default probabilities 0.5. As a result we obtain a *probability map* similar to back-projected image in [Bra98], see Figure 2. Every entry of the map keeps the face presence probability at a point of the next frame I_{t+1}.

The final steps correct the orientation and semi-axes of the face ellipse in the next frame:

1. Scale the ellipse, estimated so far, by a factor from the set {0.90, 0.92, …, 1.0, …, 1.08, 1.10}.
2. Compute covariance matrix $\mathbf{C}$ over the *area of* each *scaled* ellipse, $\mathbf{C} = ((M_{xx},M_{xy}),(M_{xy},M_{yy}))$,

$$M_{uv} = \frac{\sum_i p(x_i,y_i)k(x_i,y_i)(u_i-u_c)(v_i-v_c)}{\sum_i p(x_i,y_i)k(x_i,y_i)}$$

where $u,v = \{x,y\}$, (x_i,y_i) is a point inside a scaled ellipse, (x_c,y_c) is an ellipse center, $p(x_i,y_i)$ is the value of probability map entry (3) and $k(x_i,y_i)$ is a Epanechnikov kernel employed in [Com03] that falls down to zero at the elliptic boundary of the face (we use the similar kernel in the main algorithm).

3. Estimate new ellipse parameters as it was done in [Bra98]. To prevent new ellipse from shrinkage, we have to "normalize" its semi-axes (next step).
4. Compute another covariance matrix $\mathbf{C}_1$ assuming $p(x_i,y_i) = 1$. Uniform probability map means no change in ellipse geometry should be done and the product of eigen-values of $\mathbf{C}_1$ must be equal to the squared product of ellipse's semi-axes a and b (which are eigen-values in fact): $s^2 \det(\mathbf{C}_1) = a^2b^2$, where s is a normalization factor we need to accomplish the step 3.
5. For each scaled and "normalized" ellipse compute the mean foreground probability $\langle p_f \rangle$ inside face area F_t and the mean background probability $\langle p_b \rangle$ inside background area B_t, see Figure 1. The quality (probability) of corrected estimation is defined as follows:

$$P = \sqrt{\langle p_b \rangle_{B_t} \langle p_{\bar{f}} \rangle_{F_t}} \quad \sqrt{\langle 1 \quad p_f \rangle_{B_t} \langle p_f \rangle_{F_t}}.$$

6. Pick up the scaled and "normalized" ellipse with the highest probability P.

Let us now summarize the most important points:

1) In contrast to [Bra98], we do not query points outside the (scaled) face area while computing moments M_{uv}. This protects us from outliers.
2) We trust the center position found by the mean-shift tracker, but we try to adjust the ellipse rotation and semi-axes on each new frame by direct enumeration of scale factors ([0.9…1.1]).
3) The normalization step is an important mechanism that stabilizes the ellipse nearby the most prominent blob in the probability map.
4) At the optimal ellipse location, the contrast between the face and the background regions on the probability map attains maximum.
5) Average probabilities and moments can be computed as the contour integrals, if we prepare an integral image [Vio04] from the probability map. This provides tremendous performance gain.

3. EXPERIMENTS

During experiments we used a standalone implementation of the color tracker. The face detector, proposed in [Vio04] and implemented in the OpenCV library, was invoked on the first frame to initialize the face position.

Currently we have an efficient C++ implementation of the module that computes the optimal histogram resolution and creates the probability map (about 2 ms per frame), a relatively efficient mean-shift tracker implementation (6-7 ms with 9 scale combinations) and a blob aggregator (2.5-3 ms). The typical running time is 10-12 ms per frame on Intel Pentium 2.6 GHz, 2 Gb, single thread. We are aiming to double the performance in the next version.

Our experiments confirmed that 16x16x16 histograms adopted in [Com03] have optimal resolution most of the time, see Table 1. However, even a few "difficult" cases might be crucial for the overall stability of the method, so the resolution optimization is useful.

Figure 2. The typical probability map. Each pixel keeps the probability of object presence.

The paper describes the ongoing project, though some observations could be summarized at this stage:

1. The augmented mean-shift tracker is stable, it tolerates moderate occlusion, including occlusion by the objects of the same color (e.g. hands), and it is able to recover by itself in many situations.
2. Methods based on skin detection (e.g. [Bra98]) may fail to start, if the camera settings are not appropriate, whereas our feature classifier almost always produces a dense blob.
3. We have noticed that slow face histogram updating, adopted in [Goo02], distorts rather than improves results. This is due to changing of lighting conditions and camera instability.
4. Some particular situations cannot be handled by a color tracker only. Open chest, neck or shoulders are unavoidably detected along with the face; see Figure 2, causing a drift and loss of face ellipse. The problem could be solved by employing geometric methods, e.g. [Bir98], in addition to the color tracking approach.
5. A color tracker should be considered as an auxiliary tool that quickly localizes a face in a new frame. The next step after localization should include face detection (within a limited area), or deformable template matching, e.g. [Vio04, Dor06].

	4 bins	8 bins	16 bins	32 bins
Red	4.20%	40.4%	54.5%	0.89%
Green	2.29%	2.42%	50.2%	45.1%
Blue	0.13%	1.91%	76.8%	21.1%
All	2.21%	14.9%	60.5%	22.4%

Table 1. A typical bin number distribution of the optimal histogram per color channel.

4. REFERENCES

[Bir98] Birchfield, S. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. Proc. of the IEEE Conf. on Comp. Vis. and Patt. Recog., St Barbara, California, pp.232-237, 1998.

[Bra98] Bradski, G. R. Computer vision face tracking as a component of a perceptual user interface. In Workshop on Applications of Comp. Vision, Princeton, NJ, 214-219, 1998.

[Com03] Comaniciu, D. Ramesh, V. Meer, P. Kernel-based object tracking. IEEE Trans. PAMI 25(5), pp.564-577, 2003.

[Dor06] Dornaika, F, Ahlberg, J. Model-based head and facial motion tracking. ECCV workshop on HCI, pp.221-232, 2004.

[Goo02] Nummiaro, K., Koller-Meier, E., Van Gool, L. Object Tracking with an Adaptive Color-Based Particle Filter. In Proc. Symp. for Patt. Recog. of the DAGM, pp.353-360, 2002.

[McK99] McKenna, S. J., Raja, Y., Gong, S. Tracking colour objects using adaptive mixture models. Image and Vis. Comp. 17(3-4), pp.225-231, 1999.

[Ngu02] Nguyen, H.T.,Worring, M., Boomgaard, R., Smeulders, A. W. M. Tracking nonparameterized object contours in video. IEEE Trans. Image Processing 11(9), pp.1081-1091, 2002.

[Vio04] Viola, P., Jones, M. J. Robust Real-Time Face Detection. International Journal of Computer Vision 57 (2), 137-154, 2004.

[Zha05] Zhang, T., Freedman, D. Improving Performance of Distribution Tracking through Background Mismatch. IEEE Trans. PAMI 27(2), pp.282-287, 2005.

[Akh07] Akhriev, A. Object Tracking Via Uncertainty Minimization. ISVC(2), pp.592-601, 2007.