Face and Facial Feature Tracking for Natural Human-Computer Interface

Vladimir Vezhnevets

Graphics & Media Laboratory,
Dept. of Applied Mathematics and Computer Science of Moscow State University
Moscow, Russia

Abstract

A method for face and facial features tracking in a low-resolution web camera video stream is described. The detection results are: face bounding ellipse, eyebrow line, nostrils and mouth position. The method works at reasonable framerates on a low-quality web camera on a PIII system. The implementation of the method can be used as input module for a natural human-computer interface system to provide camera-based mouse control. The techniques used are: face tracking using color, image edge maps analysis, variant of Hough transform, template matching, color-based image segmentation.

Keywords: *Face detection, Face tracking, Facial Features tracking.*

1. INTRODUCTION

Human face detection and tracking is a necessary step in many face analysis tasks - like face recognition, natural HCI systems, model-based video coding, and content-aware video compression. Although these problems are fairly easy for a human vision system, the machine vision labels them as “hard”.

This paper describes a method for face and facial features tracking, which is designed as an input module for natural Human-Computer Interface system. This task originated from a necessity of creation of means for controlling computer for the children unable to use conventional HCI means (for example, suffering from cerebral palsy). This have set the conditions and limitations for the software – it should function on an inexpensive home computer, in parallel with the regular mouse device and should be able to work with image acquired by a cheap web camera.

Despite latest advances in the field of face and facial feature tracking (see for example [1 - 3]), most proposed methods still give acceptable results only in limited set of conditions. The reason for this is mostly the high variability of the input data. Unfortunately, many issues in the problem of robust face and facial features tracking can be treated as “unsolved” in general case. The author hopes, that methods and algorithms, described in this papers will contribute to the solution of the task of automatic analysis of human face.

2. FACE TRACKING BASED ON COLOR INFORMATION

Color is a distinctive feature of human face, making color information useful for localization of a human face on static and video images. Color information allows fast processing, which is important for a tracking system that needs to run at reasonable framerates (at least 10 fps).

Nevertheless color alone does not provide enough reliable information to detect and track a face, due to noisiness of color information and possible presence skin-colored non-face objects in the images. Some additional methods are needed to analyze the results of color segmentation. Often, connected components analysis or integral projects of skin likelihood images are used for detecting and tracking of face candidates [1 - 4]. These simple methods do not take advantage of expected shape and size of the tracked face and are prone to errors in case of non-ideal color segmentation results. The algorithm described here uses skin-colored pixel grouping method described in [5] to detect face candidate position in processed frame. This method shows robust result even when a person is positioned in front of the skin-colored background.

This method of face localization was adopted for face-tracking purposes. The algorithm for face tracking contains those steps:

1. Initialization of the face ellipse;
2. Automatic training of the skin color filter;
3. Tracking, using the created color filter and initialized face position;

During the initialization step the user sets the face ellipse position and size. Using this information, the image is divided into two areas – face and non-face and is used to train color-based skin detector. The Maximum Likelihood Bayes classifier in normalized r-g chrominance colorspace is used for skin color modeling. The r-g colorspace was chosen because of its fast and simple conversion from the RGB space, which is significant for a real-time face tracking application, and due to mentioned results of skin color modeling with Bayes ML classifier [6] (the changing of colorspaces – CIELab, YCrCb, HSV, r-g made very little impact on skin detection correctness).

Figure 1: Image used for face tracker initialization

The training is performed by gathering two statistics – $p((r,g)\,|\,skin)$ and $p((r,g)\,|\,\neg skin)$, calculated from skin and non-skin histograms built from face and non-face areas of the image. The skin likelihood image is constructed by calculating this measure for each pixel of the input frame:

$$Skin(r,g) = \frac{p((r,g)\,|\,skin)}{p((r,g)\,|\,\neg skin)}$$

Which conforms to Bayes maximum likelihood criteria (assuming a priori skin and non-skin probabilities are equal). The resulting skin likelihood image (see Fig. 2) is used for face region tracking.

Figure 2: Example skin likelihood images
(note the skin-colored background)

Elliptic model, used for face region detection, is initialized near the expected face position in the processed frame and then is adapted step-by-step to fit the image data. The step of the adaptation process consists of the considering the skin pixels lying inside the ellipse of a slightly larger size, and calculating the mass center and second order moments of the pixels formation inside this larger ellipse (see Fig. 3).

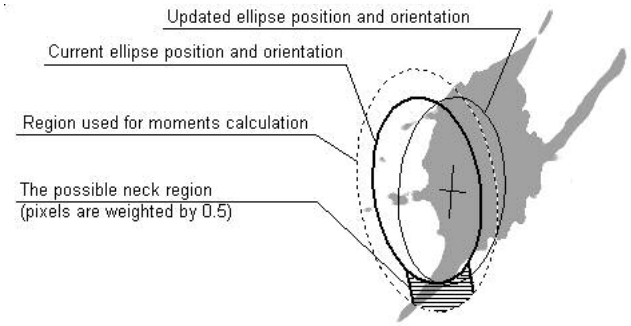

Figure 3: Elliptic model updating

Updated ellipse position and orientation is evaluated using the calculated region statistics:

μ_x, μ_y - mass center coordinates;

$\mu_{02}, \mu_{20}, \mu_{11}$ - second order central moments;

During the moments calculation, the pixels inside the minor ellipse are weighted twice against the ones in the bigger one. The pixels in the possible neck region (see Fig. 3) are weighted by 0.5 of their real likelihood, to lessen the neck influence on the face ellipse detection.

New ellipse center and axis are set at: (μ_x, μ_y) - new ellipse center point, $(\mu_{11}, \mu_{02} - \mu_{20} + \sqrt{(\mu_{02} - \mu_{20})^2 + 4\mu_{11}^2})$ - unnormalized ellipse major axis vector. Ellipse size is not updated in the current version, for the user is rarely moving from or towards the camera. Usually two or three steps of model position/orientation update procedure are sufficient for accurate face region localization (see Fig. 4).

Figure 4: Tracked faces

3. FACIAL FEATURES TRACKING

The facial features detection and tracking methods can be roughly divided into two groups: modeling features appearance with some pattern recognition method [3, 4, 8, 12], and usage of empirical rules, derived from observations of exhibited feature appearance properties [2, 9, 10, 11]. The algorithm described in this paper uses the latter idea. The features to track were chosen from the observations on the reliability of each feature tracking and also on the usefulness of the features for face orientation determination.

The minimal set of features includes: eyebrows, nostrils and mouth position.

The features detection is performed after the face region is determined. After the face ellipse is known, the face image is rotated to achieve an upright face position, to ease the process of facial features detection. Several candidate positions for each facial feature are detected (similar to [3]), then the sets of possible facial features configurations are tested (having in mind biometric rules of human face structure). Each feature's detection is based on analysis of several attributes, unique for concrete feature.

3.1 Tracking eyebrows

Usually facial feature tracking systems focus on tracking eye positions [2, 3, 4, 10, 11]. But if person wears eyeglasses, eye tracking becomes a problem, mostly because of eyeglasses highlights, so we have chosen eyebrows as a more frequently visible feature.

The detection of eyebrow line inside the face ellipse is based on the following assumptions (mind that we use a low-resolution image):

1. The eyebrows usually appear as areas significantly darker, than the forehead;
2. The forehead usually an area of smooth texture and slowly varying brightness and color;

Keeping in mind these two statements the eyebrows are found on the edge image (constructed by Prewitt edge detection operator), using a variant of Hough transform, which tries to find lines with clear areas right above them (see Fig. 5).

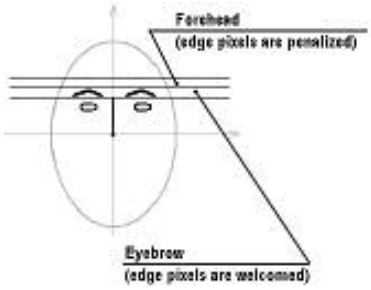

Figure 5: Eyebrow detection

The eyebrow lines are parameterized in the ellipse-based coordinate system - the center is positioned in the ellipse center and the coordinate axes are aligned with ellipse axes (see Fig. 6). The possible orientation of eyebrow line is restricted to -30 to 30 degrees from the vertical face ellipse axis (the restriction is rather vague to make correction in case of poor face ellipse orientation detection). The distance of the eyebrow line from the ellipse center is also restricted to min and max values, derived from the face ellipse size (see Fig. 7).

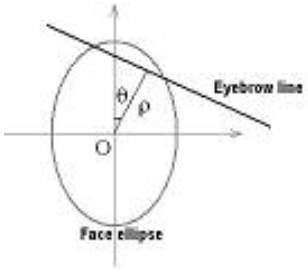

Figure 6: Eyebrow lines parameterization

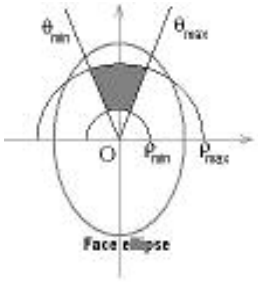

Figure 7: Eyebrow lines parameters margins

The accumulator array of dimensions $\lceil \rho_{\max} - \rho_{\min} / \Delta\rho \rceil$ by $\lceil \theta_{\max} - \theta_{\min} / \Delta\theta \rceil$, where $\lceil x \rceil$ means closest integer $\geq x$, is used for best eyebrow line selection. It is filled by the following algorithm:

For each pixel inside the upper ellipse part:

1. Transform it's coordinate to ellipse-aligned system;
2. If the pixel brightness is less than a threshold, take next pixel (goto A);
3. For each of the θ in range $(-\pi/6, \pi/6)$, with step $\Delta\theta$ calculate ρ: $\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$;
4. If ρ is in the defined margins, increment accumulator cell, corresponding to $(\lfloor \rho / \Delta\rho \rfloor, \lfloor \theta / \Delta\theta \rfloor)$, where $\lfloor x \rfloor$ means closest integer $\leq x$, and taking into account the discrete nature of accumulator array and rounding inaccuracy, also increment $(\lfloor \rho / \Delta\rho + 0.5 \rfloor, \theta / \Delta\theta)$ and $(\lfloor \rho / \Delta\rho - 0.5 \rfloor, \theta / \Delta\theta)$, in case when these cells are not same with $(\lfloor \rho / \Delta\rho \rfloor, \lfloor \theta / \Delta\theta \rfloor)$;
5. Decrement accumulator cells, corresponding to $(\lfloor (\rho - \Delta\rho) / \Delta\rho \rfloor, \lfloor \theta / \Delta\theta \rfloor)$ and $(\lfloor (\rho - 2 \cdot \Delta\rho) / \Delta\rho \rfloor, \lfloor \theta / \Delta\theta \rfloor)$ to penalize the lines, which have the bright edge pixels right above them. The neighboring cells are also decremented, if the rounding is not exact (like in 'D');

This eyebrow line detection method shows robust results, making eyebrows a stable feature to track (see Fig. 8).

Figure 8: Eyebrow detection

3.2 Tracking lips

Some methods ([3], [12]) use brightness information, or pre-trained color predicate [2] for lip detection. We have tried to use method of lip detection, which does not need prior calibration, but differentiates the lips from the face, using color information.

The lip tracking method is based on assumption that lips usually have characteristic color, differing from generic face color (similar method used in [7] and [11]);

Do detect probable lip regions several steps are taken:

- Face image is transformed to special one-channel image function:
  $Lip(RGB) = (u/v) \cdot 0.07 + (1 - Skin(r,g)) \cdot 0.3$
  where (u,v) - are the chrominance coordinates in CIELuv colorspace, $Skin(r,g)$ is the skin likelihood for the current color, and 0.07 and 0.3 are the empirically selected weights;
- Taking into account the noisiness of web-camera image and the low resolution of the picture, some filtering of the "lip function image" for noise elimination is performed:
  - max filter with 3x3 window;
  - median filter with 3x3 window;
  - thresholding with 0,4 threshold value;
  - morphological opening with 5x5 round mask;

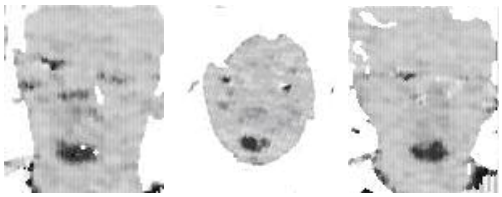

Figure 9: Lip function images

To find the region, corresponding to lips (it can be seen, that some spurious high "lip function" values exist - see Fig. 9), an elliptic template is applied to the lip region, to find most probable lip position. The template's size is chosen proportional to the face ellipse size. The templates goodness of fit is measured by:

$$E(x_c, y_c) = \alpha \cdot \sum_{(x,y) \in In(x_c, y_c)} Lip(x,y) - \beta \cdot \sum_{(x,y) \in Out(x_c, y_c)} Lip(x,y);$$

where $In(x_c, y_c)$ - is the inner area of the template positioned at (x_c, y_c) and $Out(x_c, y_c)$ - is the border and part of template outer area (marked with light gray in the figure), and $Lip(x,y)$ is the "lip function" value in the image location (x,y).

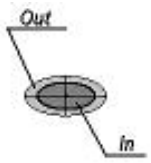

Figure 10: Lip template with marked *In* and *Out* regions

To find most probable lips center the positions, which give not less than 90 % of the maximum goodness of fit are averaged. The results of lips detection are showed in the figure. Of course, the assumption about the vivid lips color, unfortunately, is not always true. The cases, when this assumption fails will be addressed in the future research.

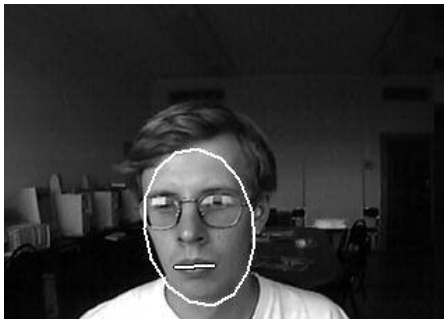

Figure 11: Detected lips

3.3 Tracking nostrils

If a camera is positioned properly, nostrils are very clearly visible at most head orientation angles, giving a stable and relatively easy-detected feature. They represent two high contrast regions, exhibiting certain brightness patterns (dark round spot inside a relatively bright background) and high brightness gradient values (see Fig. 12).

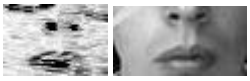

Figure 12: Nostril area on gradient and grayscale images

The nostrils detection is performed by scanning the part of face area with templates, which try to find regions that exhibit high brightness gradient values and low brightness in the middle, while showing high brightness values at the region borders. Median filter is applied to the grayscale image, prior to nostrils detection. The most likely positions form a set of nostril candidates, which are then tested relatively to face position, detected eyebrows and each other to find the pair of the most probable nostrils positions.

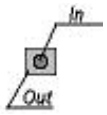

Figure 13: Nostril template with marked *In* and *Out* regions**;**

The nostril template goodness of fit is measured by the formula:

$$E(x_c,y_c) = \alpha \cdot \sum_{(x,y)\in In(x_c,y_c)} I(x,y) - \beta \cdot \sum_{(x,y)\in Out(x_c,y_c)} I(x,y) + \gamma \cdot \sum_{(x,y)\in In(x_c,y_c)} \left|\nabla I(x,y)\right|$$

where *In* and *Out* areas of the nostril template are shown in the figure, and $I(x,y)$ and $\left|\nabla I(x,y)\right|$ are the image brightness and absolute value of the brightness gradient vector respectively.

The examples of the detected nostrils are shown in the figure:

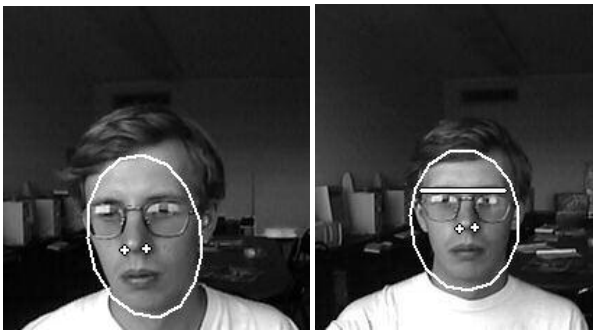

Figure 14: Detected nostrils

4. CONCLUSION AND FUTURE WORK

A face and facial features tracking method is described, which works at reasonable framerates on a low-quality web camera on a PIII system. The most reliable features, as the experience shows, are the eyebrow line and the nostrils positions. The implementation of the method can be used as a module for a natural human-computer interface system, for the information provided by the algorithm (face orientation and facial features positions on each frame) can be used for camera-based mouse control.

The currently used methods, although showing acceptable performance in most cases, leave a plenty of room for improvement: first of all, this concerns the limitations and assumptions made in each case. The increasing of robustness and accuracy as well as tracking other facial features is another issue. Also, as the face is known as a highly variable structure, which shows very different appearance from image to image, more intelligent selection of feature candidates, analyzing and predicting the features positions in the whole may be helpful. Each feature can be assigned a confidence measure based upon pre-defined information (lighting conditions, presence of eyeglasses and facial hair) and tracking error statistics, gathered during the system functioning. The system may use different algorithms of feature candidates' detection and verification, depending of these confidence values.

5. REFERENCES

[1] K. Schwerdt and J. Crowley, "Robust Face Tracking using Color", *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 26 - 30 March, 2000, Grenoble, France*

[2] Paul Smith, Mubarak Shah, and Niels da Vitoria Lobo, "Monitoring Head/Eye Motion for Driver Alertness with One Camera", *Fifteenth IEEE International Conference on Pattern Recognition, September 3-8, 2000.Barcelona, Spain*

[3] Alper Yilmaz, Mubarak A. Shah "Automatic Feature Detection and Pose Recovery for Faces", *ACCV2002: The 5th Asian Conference on Computer Vision, 23--25 January 2002, Melbourne, Australia*

[4] S. Spors, R.Rabenstein "A Real-Time Face Tracker For Color Video" *IEEE Int.Conf. on Acoustics, Speech & Signal Processing (ICASSP)*, Utah, USA, May 2001

[5] V. Vezhnevets "Method For Localization Of Human Faces In Color-Based Face Detectors And Trackers" *The Third International Conference on Digital Information Processing And Control In Extreme Situations*, May 28-30, 2002, Minsk, Belarus.

[6] B. D. Zarit, B. J. Super, and F. K. H. Quek, "Comparison of five color models in skin pixel classification". In *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 58-63, Kerkyra, Greece, September 1999.

[7] S. Soldatov "Lip reading: Lip contours detection", *Intellectual Information Processing Conference*, 17 – 21 June, 2002, Alushta, Russia (in russian)

[8] A. Colmeranez, B. Frey, Th. S. Huang “Detection and Tracking of Faces and Facial Features”, *ICIP 1999*, pp.657-661, 1999.

[9] V. Bakic and G. Stockman, “Menu Selection by Facial Aspect”, Proceedings of Vision Interface 99, Trois Rivieres, Quebec, CAN (19-21 May 99).

[10] K. Toyama, "'Look, Ma -- No Hands!' Hands-Free Cursor Control with Real-Time 3D Face Tracking", *In Proc. Workshop on Perceptual User Interfaces (PUI'98)*, San Francisco, November 1998

[11] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, ``Face detection in color images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696-706, May 2002

[12] M. Gargesha and S. Panchanathan, "A Hybrid Technique for Facial Feature Point Detection", *Fifth IEEE Southwest Symposium on Image Analysis and Interpretation*, 7 - 9 April 2002, Santa Fe, New Mexico

About the author

Vladimir Vezhnevets is the PhD student of department of Applied Mathematics and Computer Science of Moscow State University.

E-mail: **vvp@graphics.cs.msu.su**