Active Face and Feature Tracking *

Luis Jordão Matteo Perrone João Paulo Costeira
José Santos-Victor
Instituto de Sistemas e Robótica - Instituto Superior Técnico,
Av. Rovisco Pais, 1049-001 Lisboa Codex, PORTUGAL
{jordao,surf,jpc,jasv}@isr.ist.utl.pt

Abstract

This paper describes a method for the detection and tracking of human face and facial features. Skin segmentation is learnt from samples of an image. After detecting a moving object , the corresponding area is searched for clusters of pixels with a known distribution. Since we only use the hue (color) component this process is quite insensitive to illumination changes. The face localization procedure looks for areas in the segmented area which resemble a head. Using simple heuristics, the located head is searched and its centroid is fed back to a camera motion control algorithm which tries to keep the face centered in the image using a pan-tilt camera unit. Furthermore the system is capable of tracking, in every frame, the three main features of a human face. Since precise eye location is computationally intensive, an eye and mouth locator using fast morphological and linear filters is developed. This allows for frame-by-frame checking, which reduces the probability of tracking a non basis feature, yielding a higher success ratio. Velocity and robustness are the main advantages of this fast facial feature detector.

1 Introduction

The work described in this paper is part of an ESPRIT project called VIRSBS which is an acronym for Visual Intelligent Recognition for Secure Banking Services [13]. The main goal is to provide a new security level in ATM (Automatic Teller Machine) using vision. The project is divided in four main tasks, person detection and tracking of a face, rough estimation of features localization (eyes and mouth), precise feature localization and extraction and the verification. In this paper we describe the first two modules, the person detector and the face and feature tracker.

New methods were developed to deal with skin, head and face features localization. These methods were developed keeping in mind an ATM interaction where the objective is to confirm that a person's identity is the one on their user card, using information provided by vision [14]. One of main goals is to design a robust system requiring the least cooperation from the user. In other words, the user should not be constrained to fixed positions, pose or movements. To cope with this, we used a pan-tilt camera unit which tracks the user while he moves towards the camera and tries to keep the face always centered in the image. Such an unconstrained environment poses huge requirements on speed, therefore the face and eye/mouth detectors should be robust but fast.

We use color information to segment skin areas. Post processing decides which areas are likely to be a face. This image region is tracked by a moving camera so that the pattern is always centered in the image. Controlling the camera in velocity we were able to track the face with a smooth but fast motion at low frame rate.

Also a new method for fast feature detection of facial characteristics is presented. The use of non-linear morphological operators allows us to select, within a close-up of a face, the three strongest features for classification. This approach allows a fast but not so precise eye and mouth localization. The estimated feature positions are fed back to a precise eye locator, which runs after card insertion, and described elsewhere [10]. Repeating this feature selection over time, and treating all the images separately, permits the detection of incorrect feature selection frames, implicitly giving more robustness to the system.

2 Using color for face segmentation

In recent years, extensive research has been carried out on automatic face localization. Some of the proposed methods are based on correlation of fixed templates or variations such as eigenface matching techniques [8] while others are based on deformable models, such as the approach purposed by Lanitis [5]. Other techniques based on HMM

*This research was partially funded by project ESPRIT-LTR-21894

(Hidden Markov Models) [9], neural nets [12] and genetic algorithms [6] have also been proposed. However all these methods are computationally expensive. Recently, in order to reduce complexity and increase speed, color has been extensively used for face segmentation.

Color is one of the most important characteristics of the human face and is often used as a method to locate a face on images [11] [3] [15] [7]. These methods use techniques such as Kalman filter, region growing or color correlation that are computationally demanding.

2.1 Color space

The most common way of representing color is through the RGB color space. However this color model is quite sensitive to lighting conditions since the color attribute is combined with the brightness one. There is a group of color spaces that have separated information for brightness and color. HSV (Hue Saturation Value) color space is the one we chose. It represents a color image with three components: Hue (color frequency), Saturation (how "pure" the color is) and Value (brightness).

The HSV model can be obtained from RGB as follows:

$$H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B)}} \quad (1)$$

$$S = 1 - 3\frac{min(R,G,B)}{R+G+B} \quad (2)$$

$$V = 0.299R + 0.587G + 0.114B \quad (3)$$

From the statistical analysis of large data sets we concluded that the hue distribution of skin is approximately gaussian with mean angle $\bar{\theta} \approx 0°$ and a variance $\sigma \approx 7°$. Therefore skin areas can be easily detected by thresholding the hue component of an image using as thresholds, for example, $T_{low} = \bar{\theta} - 2\sigma$ and $T_{high} = \bar{\theta} + 2\sigma$, as shown in Figure 1.

Even people of different races have skin color with (approximate) the same dominant frequency in the color spectrum (hue), but different spread (saturation) [1] [15]. Furthermore the hue variance is small and fixed.

2.2 Auto-Calibration

In a well-known environment the procedure described above is sufficient to detect skin color but several problems can arise if the system has to work in a completely unconstrained environment. For example if the camera is exposed to colored light(sun light) or if another camera is used (with different color parameters or different acquisition hardware settings) the results will change. Analyzing the behavior of color in these conditions, we concluded that the mean value of the hue distribution changes but the variance is preserved.

In order to avoid this fact and introduce some robustness, we developed an adaptive mechanism where the system automatically tunes up the threshold parameters needed for segmentation. We called this the auto-calibration procedure. In order to avoid incorrect target detection we use motion as a trigger procedure. The auto-calibration process is only started when and where motion is detected. Keeping in mind the particular application, the person has to approach the camera and therefore must generate image motion.

In the area where motion is detected a grid is applied. For each window of the grid the mean and variance of hue distribution is computed. Using previous knowledge of the skin-color distribution each window can be classified as a skin color candidate or not, as shown in Figure 2 (a).

Analyzing all of the window candidates we choose the one which best complies with the prior and we set the thresholds used in segmentation in the same way as referred to before. If no candidate is found, the motion was caused by something that was a not a person and the process restarts.

3 Face Localization

Now that we have the segmented (binary) image, we need to find the face. In order to have a fast algorithm we use the summing projections of the segmented image $I_{seg}(N \times M)$ (Figure 1 (b)) in the vertical and horizontal axis [2].

$$proj_{hor}(x) = \sum_{y=0}^{N} I_{seg}(x,y) \quad (4)$$

$$proj_{ver}(y) = \sum_{x=0}^{M} I_{seg}(x,y) \quad (5)$$

Using this tool we work in two 1D signals which is much

(a) (b)

Figure 1. Skin color detection (a) Original image (b) Segmented image.

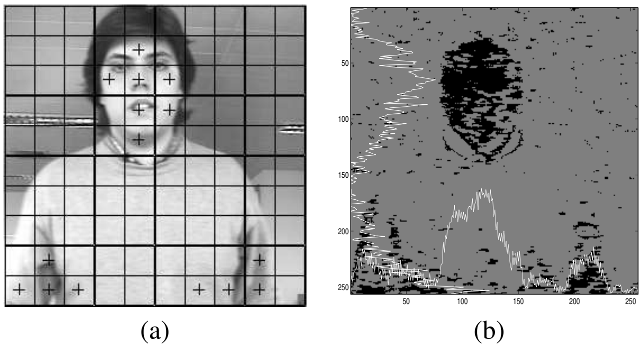


Figure 2. (a) Skin color candidates (the cross identifies possible skin color areas) (b) Vertical and horizontal projections of segmented image.

faster than working in the image (2D), as shown in Figure 2 (b).

To find the face in the image we start analyzing the horizontal projection. The median position of the integral area of the projection will be approximately the center of the "biggest" skin color blob in the image. To find the boundaries of the blob, we analyze the signal in both directions starting from the median position. In an ideal image, non-blob areas will give a zero signal, but since we have a noisy hue image we use a threshold that depends on the width of the image. After this procedure we have a vertical slice of the image where the face could be. In this sub-image the same procedure is applied to the vertical projection. The final result will be a window surrounding the face.

Since we could have very noisy images and we want to track the center of the face we compute the centroid of segmented image inside the window where the face is located. As can be seen in Figure 3 the information supplied by the system is the center of the face and the dimensions of the window surrounding the face.

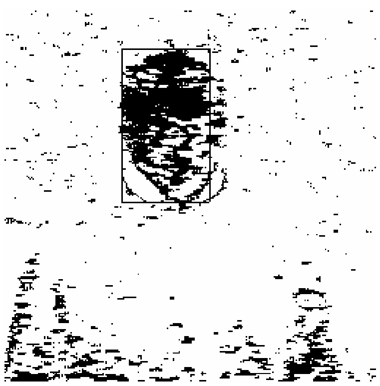


Figure 3. Face located.

3.1 Multiple Blobs

We have another problem to solve, in that the image could contain several faces or the hands of a person. By generalizing the previous procedure it is possible to detect multiple blobs (Figure 4) and we have to determine which blob is indeed the face. For the sake of speed we have selected the "biggest" blob. This simple heuristic is enough to detect the face we are interested in. If there are several people in the image we choose the person that is closer to the camera (the biggest blob).

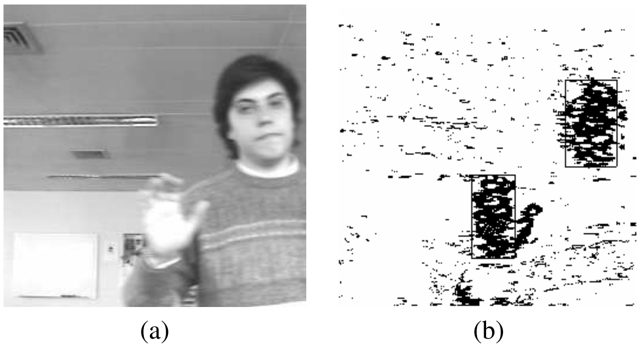


Figure 4. Multiple blobs (a) Original image (b) Detected blobs.

4 Camera Control

Since we are dealing with a dynamic environment where we must fixate on a person's face we use a pan-tilt controlled camera which keeps the face in the center of the image, as shown in Figure 5. The camera pan/tilt velocities are computed by a closed loop controller driven by the image position of the face over consecutive frames.

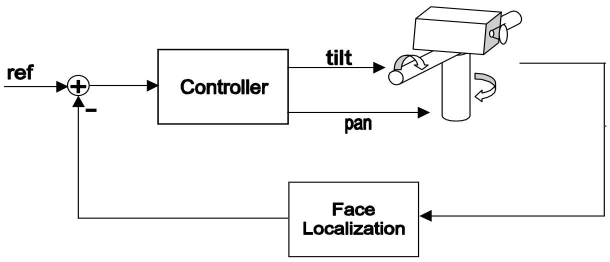


Figure 5. Control scheme.

Since people have, in general, smooth motion, the controller sets the velocity in such a way that, if the target keeps moving at the same speed, the face will be centered. This allows reasonable tracking at much lower sample rates. To test the controller robustness to sudden changes in velocity, we tracked a person waving from left to right located two meters far from the camera. Figure 6 shows the face image location. The sample rate was 4 frames per second.

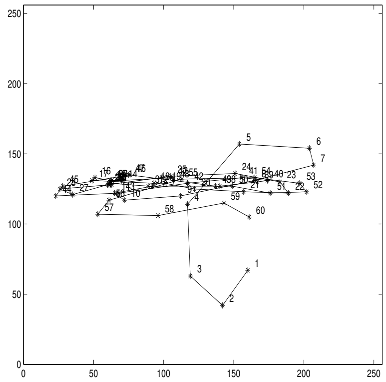


Figure 6. Tracking a person.

At such a low frame rate we were able to guarantee that the face was located about the center of the image. At each frame our controller always kept the face within the image boundaries.

5 Finding facial features

Given the dimension and the position of the face locating window in the image, as shown in the previous sections, it is possible to detect basic facial features in a qualitative way. This method uses morphological filters to enhance the spatial frequency yielded by the eyes, nose and mouth. This allows their location in a very fast and robust way. The main characteristic of this algorithm is its high flexibility which yields successful results, even in the case of facial expression and pose changes, or in presence of extra features such as beards, mustaches or glasses.

For every frame two operations occur: generation of the filtered image, and main facial features detection. The former is generated by a sequence of morphologic operations on the extracted window. The computed image is given by $G = h \star I_t$ where h is a low pass filter and I_t is:

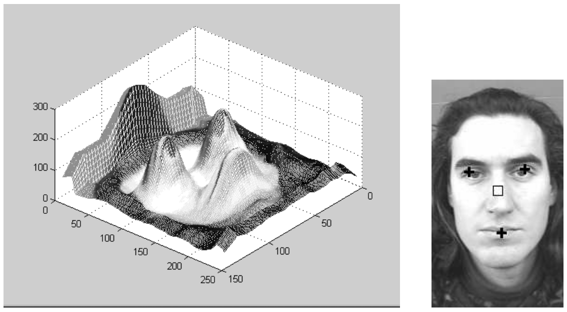
$$I_t = |I - open(I)| + |I - close(I)| \qquad (6)$$

where I is the extracted window and $open(I)$ and $close(I)$ are the gray-scale morphological open and close operators [4]. Note that $|I - open(I)|$ enhances peaks in the image and $|I - close(I)|$ enhances valleys. Figure 7 shows the face of a subject, and the result of the computation.

Clearly, the filtered image contains only relevant face information. Figure 8 (a) shows the original gray level image

Figure 7. From left to right: Original image I, Peaks, Valleys, Sum of Peaks and Valleys I_t and Filtered image G

mapped onto the filtered image G. As can be seen, the preprocessing yields a stable image, and it shows the deviation in value between points of interest and others. Since face tracking provides a raw estimate of scale, finding the first N local maxima of G, we compute P triangles which are combinations of three maxima. The feature locations are given by the triangle that best fits the size constraints. Figure 8 (b) shows the detected features.



(a) (b)

Figure 8. (a) Computed image elevation (b) detected features. The square in the middle represent the center of mass for face tracking.

Moreover, the fast computation allows for frame by frame feature tracking thus conferring increased reliability on the system. An incorrectly recognized feature in the first few frames will not adversely affect the results since the system does not require memory to predict new possible positions. It will find them without previous information yielding a robust error detector.

6 Conclusions

In this paper we have presented a novel method for fast and robust, face and facial feature finding. The method can be divided in two phases. In the first place a human face is detected and tracked in the scene. To achieve this, color

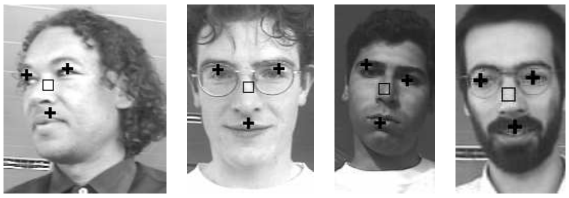


Figure 9. Successful detections. As drawn in these examples, features are localized at reasonable distances from the correct location regardless of artifacts like glasses, beards, mustaches or skin color.

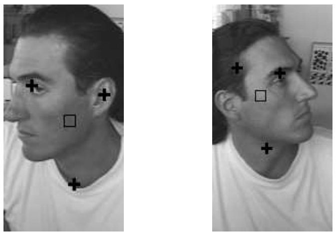


Figure 10. Failures due to strong pose

information was used to segment skin. The face localization procedure looked for a segmented area that could be a head. The center of mass of this area is tracked over time using a velocity controlled pan-tilt camera unity. This control scheme provides fast but smooth motion at low frame rate. Frame by frame a window locating the face is extracted and passed to the facial feature procedure. The use of morphologic operators allows an on-line adjustable image filtering able to enhance all the possible candidates of facial features inside a window extracting a human face in the scene. Subsequently, a geometric model based on face morphology, detect among the group of candidates the triplet that better locates eyes and nose-mouth. Repeating this procedure over time, we have a frame by frame features detector that, not depend on the previous solutions, is capable at pointing out the bad recognized features thus avoiding the use of invalid frames. Therefore we obtain a more precise feature localization, or identity recognition/validation of the subject.

Current experiments are being done to compute precise statistics but the system has been tested at a 4 day show at IST98 (Information Society and Technology) in Vienna without failures.

References

[1] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal Q2 '98*, 1998.

[2] R. Brunelli and T. Poggio. Face recognition through geometrical features. In *Proc. of European Conference on Computer Vision*, S. Margherita Ligure, 1992.

[3] P. Fieguth and D. Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[4] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall International, Inc., 1989.

[5] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic tracking, coding and reconstruction of human faces, using flexible appearance models. *Electronic Letters*, 30:1578–1579, 1994.

[6] H. Nugroho, S. Takahashi, Y. Ooi, and S. Ozawa. Detecting human face from monocular image sequences by genetic algorithms. In *Proc. of ICASSP*, 1997.

[7] N. Oliver, A. P. Pentland, and F. Berard. Lafter: Lips and face real time tracker. In *Proc. of Conference on Computer Vision and Pattern Recognition*, S. Juan, Puerto Rico, 1997.

[8] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle USA, July 1994.

[9] A. N. Rajagopalan, K. S. Kumar, J. Karlekar, R. Manivasakan, M. M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri. Finding faces in photographs. In *Proc. of International Conference on Computer Vision*, Bombay, India, January 1998.

[10] F. Smeraldi, N. Capdevielle, and J. Bigun. Facial features detection by saccadic exploration of the gabor decomposition and support vector machines. In *Proc. 11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 1999.

[11] K. Sobottka and I. Pitas. Segmentation and tracking of faces in color images. In *Proc. of the Second International Conference on Automatic Face and Gesture Recognition*, 1996.

[12] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.

[13] M. Tistarelli, E. Grosso, J. Bigun, C. Sacerdoti, J. Santos-Victor, and D. Vernon. The virsbs project: Visual intelligent recognition for secure banking services. In *Proc. of the 9th International Conference on Image Analysis and Processing*, Florence Italy, September 1997.

[14] M. Tistarelli, A. Lagorio, M. Jentile, and E.Grosso. Design of a vision system for identity verification. In *Proc. of HICSS-32*, Honolulu Hawaii USA, January 1997.

[15] J. Wang and A. Waibel. A real-time face tracker. In *Proc. of WACV '92*, Sarasota Florida USA, 1996.