MYRA – Face Detection and Face Recognition System

Tibor Kövér, Dénes Vígh, Zoltán Vámossy

Institute of Software Technology, John von Neumann Faculty of Informatics, Budapest Tech
Bécsi út 96/B, H-1034 Budapest, Hungary
Phone: +36-1-219-6550, Fax: +36 –1-219-6551
E-mail: vamossy.zoltan@nik.bmf.hu

Abstract: MYRA is a system created for face detection and face recognition on video images or live webcam video stream. For face detection the program uses two very different parallel techniques to give a better performance and for recognition it uses the eigenfaces method. The system includes a small database that stores information and optionally some personal data about the people already known by the system.

Keywords: face detection, face recognition, eigenfaces, skin tone detection

1 Introduction and Aims

The two main goals of the project are face detection and face recognition. The source of the information is a video camera or a webcam. Face detection means that the program is able to find human faces on a video stream, also able to track them and tries to determine their orientation. To make the system more reliable it is necessary to use not only one, but more different techniques that are different enough to supplement one another. There is a system developed earlier in our institute [1] that used the Hough transformation for finding faces, based on the presumption that the human face appears as an oval shape on the image. Now this system uses two different techniques, an appearance based method and skin tone detection. Since they work in quite a different way, it can be assumed that they complement each other well. Another important task relating to face detection is face tracking, which means that once a face is found on the video stream the system sticks to it and tries to follow that face even if its orientation or illumination changes in such a way that the two face detection techniques mentioned above can not find it on the subsequent frames.

The second main goal of the project is face recognition. Of course there is a prerequisite for this, the face has to be close enough and be in a good position related to the camera for the system to be able to recognise it. The system is expected to recognise those people once it has seen (and has been able to extract enough features for recognition). So if a person known by the system steps into the view of the camera, then the program has to recognise him or her and supply information stored in the system database about the given person.

2 Face Detection and Recognition Systems

As computer technology and image processing has undergone a rapid development at the end of the twentieth century face recognition systems started to appear one after the other. The reason is simple, the performance of computers grew at a very fast pace while they became cheaper and cheaper, so did analogue and digital cameras too. Commercial products and application kits appeared and they started to be used in airports, banks, stadiums, ATMs, shops and public facilities. Their aim is mainly processing videos recorded by security cameras and police uses similar systems for searching in large face databases. Sources [2] and [3] give detailed summary about face detection systems and applied methods.

One such system is the FaceSnap (www.facesnap.de), which is a complex, expandable application kit for face detection and face recognition using neural networks. It has several modules like FaceSnap RECORDER that was developed for security purposes and is used in airports and banks. Its task is to evaluate the videos recorded by security cameras. The FaceSnap FOTOMODUL helps to build up a face database; it founds faces and creates normalized photos about them. The Face Check Verify module works as a biometric access control system.

3 The MYRA System

3.1 Face Detection

The Myra System uses two different techniques for finding faces, an appearance based method and a skin tone detection method. These two techniques are meant to make face detection more accurate, so that on the one hand it would be less sensitive for changes of the illumination, on the other hand it should find faces even if the face on the picture does not look directly into the camera, so when the appearance based method can not be applied (which is able to detect frontal or profile faces, but can not cope with those that are between the two pose). The two methods can also be used to check the result of each other.

The appearance based method was implemented according to the method described in source [4], for this purpose the function of the OpenCV library was used that can give very good, accurate results in real time if the direction of the face is frontal on the images. However, since the parameters of the function are set to support robust processing and so are changed in a way to help speeding up the whole system, it may give more false positive alarms than it would do with default values. So the accuracy of the function is traded off to support robustness and keep up the continuity of the live video stream. For this reason one main task of the other face detection method is to filter out these false positive results.

The other basis of face detection in the MYRA system is a skin tone detection algorithm [5], which was supplemented with some of our own methods, pre- and post-processing procedures to improve the performance of the detection. Prefiltering mainly consists of a noise filtering algorithm and some color temperature correction, while post filtering is the correction of the skin tone map.

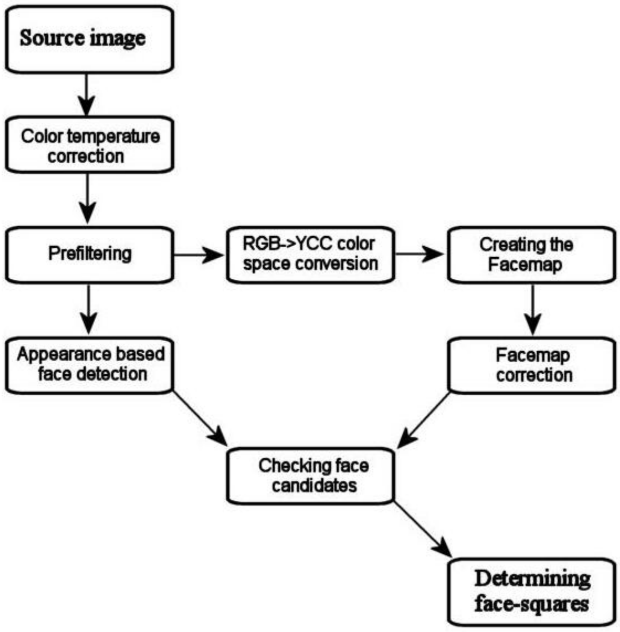

Figure 1

The flowchart of the face detection in MYRA

Figure 1 shows the main steps of MYRA System’s face detection method, the way they are connected and the execution sequence. First of all a color temperature correction is carried out. Its purpose is to improve the white balance of the overall image. The next step is a prefiltering process, namely a blur filter. Though the program uses blur as a default, this filter can be changed from the program in a quite simple way, gauss and median filters are also available and can be used. Blur is default because this is the fastest and still gives a fair good result in smoothing the image, but median would do the best job.

After prefiltering the resulting image is used and processed by the two different face detection methods. One is the appearance based technique, which is carried out with the help of the Adaboost method and Haar-like filters [4, 6]. It is implemented using the above mentioned functions and procedures of the OpenCV library. Training samples in good quality are also available for them.

Following the other branch of the detection a non-linear color space conversion is carried out as the first step of skin tone detection. It is necessary, since the method uses the YCC color space and not the RGB. The great advantage of the YCC color space is that illumination and the real color (luminance and chrominance) of each pixel are separated into different channels, and thus the color is much less dependant on the illumination, not like in the RGB color space.

Conversion is non-linear only because this way it is much easier to determine a compact area where skin tone color pixels can be found. The equations used to achieve this conversion are as follows (according to [5]):

$$C'_i(Y) = \begin{cases} \left(C_i(Y) - \overline{C}_i(Y)\right) \cdot \dfrac{W_{Ci}}{W_{Ci}(Y)} + \overline{C}_i(K_h) & \text{if } Y < K_l \text{ or } K_h < Y, \\ C_i(Y) & \text{if } Y \in [K_l, K_h], \end{cases}$$

$$W_{Ci}(Y) = \begin{cases} WL_{Ci} + \dfrac{(Y - Y_{\min}) \cdot (W_{Ci} - WL_{Ci})}{K_l - Y_{\min}} & \text{if } Y < K_l, \\ WH_{Ci} + \dfrac{(Y_{\max} - Y) \cdot (W_{Ci} - WH_{Ci})}{Y_{\max} - K_h} & \text{if } K_h < Y, \end{cases}$$

$$\overline{C}_b(Y) = \begin{cases} 108 + \dfrac{(K_l - Y) \cdot (118 - 108)}{K_l - Y_{\min}} & \text{if } Y < K_l, \\ 108 + \dfrac{(Y - K_h) \cdot (118 - 108)}{Y_{\max} - K_h} & \text{if } K_h < Y, \end{cases}$$

$$\overline{C}_r(Y) = \begin{cases} 154 + \dfrac{(K_l - Y) \cdot (154 - 144)}{K_l - Y_{\min}} & \text{if } Y < K_l, \\ 154 + \dfrac{(Y - K_h) \cdot (154 - 132)}{Y_{\max} - K_h} & \text{if } K_h < Y, \end{cases} \quad (1)$$

The base for this method is the fact that in the YC_bC_r color space chrominance can be written as function of the luminance. That way the transformed chroma is made up of $C_b'(Y)$ and $C_r'(Y)$. In the first two equations C_b-t or C_r has to be substituted in place of C_i. W_{cb}, W_{cr}, WL_{cb}, WL_{cr}, K_l, K_h, Y_{min} and Y_{max} are predefined constants, whose values are derived from practise.

After the color space conversion is done skin tone color pixels are to be found on a definite area of the 2D projection of channels C_b and C_r. It is approximately the shape of an ellipse, so the equations that are used to determine if the given C_b - C_r values are within this area are derived from the equations of an ellipse.

$$\frac{(x-ec_z)^2}{a^2}+\frac{(y-ec_y)^2}{b^2}=1$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} C'_b - c_x \\ C'_r - c_y \end{bmatrix} \tag{2}$$

This system of equations also have quite a few constants, for example c_x, c_y, ec_x, ec_y, *theta*, *a* and *b*. The ellipse defined by the two equations above can be seen on figure 2, which shows the C_r - C_b projection of the color space. The cohering grey blob is the cloud of skin tone color pixels determined by experience.

Skin tone color pixels found by the technique described above are stored in a 320x240 matrix (the exact size of one video image frame). This matrix is binary, so it can have only two elements in it, 1 and 0. This matrix will be referred as the skin map from now on.

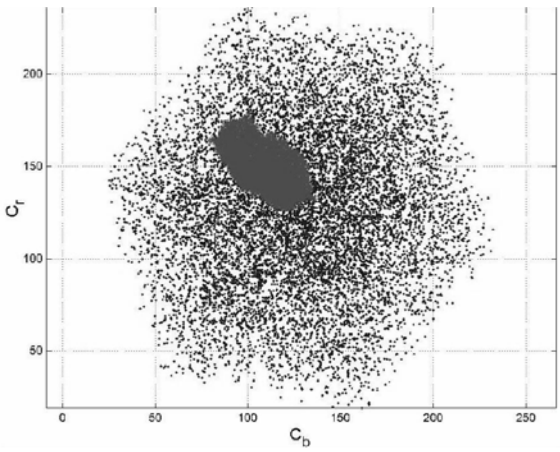

Figure 2

The C_b-C_r projection of the YCC color space having the skin tone color ellipse in the middle [5]

After all these computations one thing is left, that is the correction of the skin map. Its purpose is eliminating lonely skin color pixels and smaller pixel groups from the map, while filling in the gaps of the larger skin tone areas at the same time. Basically there are two solutions for this. The first (also the default) is a simple method called closing, which is an erosion and a dilatation, one after the other in this order. Erosion eliminates lonely pixels, and then dilation fills in the little gaps. Since this technique results in only very modest changes, there is an alternative for this, an other method which can make greater changes and gives a much better result. This is our novel heuristic technique that is basically a sequence of an erosion (with a 5x5 mask) and three dilations (with a mask of 3x3). The erosion here has special rules; the 25 pixels covered by the mask are divided into frame and inner pixels. If the frame has at least 6 skin tone pixels and the inner part has 7 OR the frame has at least 10 skin tone color pixels then the pixel in the middle of the mask is regarded as a skin tone color pixel further on, regardless of what value it had before. If the above conditions do not stand, then the pixel gets the value of the background.

After the skin map is created as described above, the results of the two different techniques are compared and combined. It has to be examined whether the face candidates found by the appearance based method contain a large amount of skin tone pixels (as it could be expected) or not. If so, then the candidate can be regarded as a face found on the picture, else the face candidate has to be rejected and removed from the list. For this step the system uses a threshold value that can also be set manually. The area of a face candidate has to contain at least 70% skin tone pixels to be a face as a default, but this threshold might be set to a much lower value (40-50) still giving a good and reliable performance.

There is one more essential task that belongs to the face detection part and that is face-tracking, where the computation speed optimization is important. The main goal here is to spare some time (the face detection algorithm should not have to be applied to every single frame), and face-tracking also helps the program to see those faces that are not in an ideal position for face detection. This tracking method also uses the skin map, because even if a face is turned or rolled the cloud of skin tone pixels will remain the same place. So if the appearance based method loses the face found on the previous frames, but the skin tone pixels are still there, then the program has a good reason to think that the face is still there and will show it, improving the continuity of the detection. In these cases the program works in a predictive mode and if certain conditions stand, than it will continue to show the faces found earlier on the place they are supposed to be. If we know that the position of the camera is fixed and it does not move or shake, then we can also use a method to subtract the background from the moving foreground, this way helping the work of the algorithm.

3.2 Face Recognition

The second main function of the Myra system is face recognition. It uses the “eigenfaces” method [8], which was developed by Kirby and Sirovich at Brown University, including its developments in the meantime. The eigenfaces method is an effective and uncomplicated procedure for face recognition based on efficiently extracting the relevant information from human face images. The relevant information is extracted by calculating the difference between the actual image and an “average”, mean face image. In other words, the main (principal) components of the face images are sought. This procedure is called Principal Components Analysis (PCA) and it is an appropriate method for the evaluation of data sets because it shifts out their differences and equalities.

In the first step, the average image is computed from the training image set.

$$S = \{ \Gamma_1, \Gamma_2, \Gamma_3, \ \ldots\ldots\ldots \ , \Gamma_M \}$$

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n \tag{3}$$

Then using the average image, difference images are constructed by subtracting the training images from the average image.

$$\Phi_i = \Gamma_i - \Psi \tag{4}$$

Then the difference images are transformed in form of vectors into multidimensional space. These vectors, all representing faces, are not randomly located in space. They concentrate in a certain region, building a cloud. This part of space is called the facespace. The facespace can be described by eigenvectors and therefore a smaller number of dimensions is needed than for the description of the whole space.

The eigenvectors are the eigenfaces that carry the most characteristic differences from the average image. In order to compute the base of the image space the eigenvectors and eigenvalues of the covariance matrix *C* must be calculated which is composed of the difference vectors of the tutor set.

$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \\ &= AA^T \end{aligned} \tag{5}$$

Trying to calculate the eigenvectors of the huge n x n matrix C would exceed the limitations of present computing equipments, that is why the Turk method is used.

$$L_{mn} = \Phi_m^T \Phi_n \tag{6}$$

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k \qquad l = 1, \ldots\ldots \ , M \tag{7}$$

The point of the Turk method [9] is that instead of calculating the eigenvectors, describing the space of eigenfaces, by computing them from the covariance matrix *C*, the eigenvectors of the matrix m x m (m << n) are calculated, that is by magnitudes smaller and it is proved to contain the eigenvectors of *C*.

Establishing the eigenvalues and eigenvectors was a programming assignment because the Delphi environment does not contain such routines.

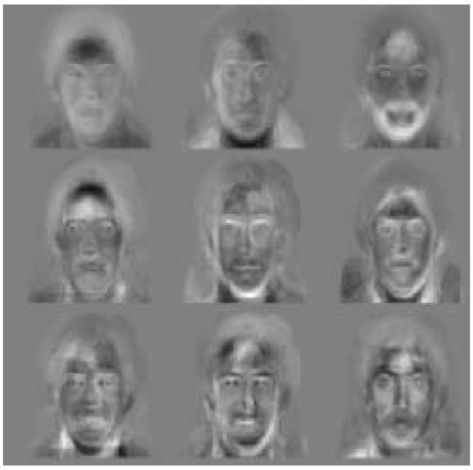

Figure 3
Eigenfaces [9]

To calculate the eigenvectors and eigenvalues the Jacobi rotation algorithm [10] is used. This is a sequence of iterations to rotate the symmetric matrix in order to get its eigenvector matrix (a covariance is a symmetric matrix).

The first step of recognition is to transform the input image into the facespace. This means to represent the input image as a linear combination of the eigenfaces' base vectors.

$$\omega_k = u_k^T (\Gamma - \Psi) \tag{8}$$

The resulting vector will be compared with the other stored image vectors from the database during the process of recognition. The best resemblance is achieved when the Euclidian distance of the weighted vector of the stored image is the smallest to the weighted vector of our image.

$$\varepsilon_k = \|\Omega - \Omega_k\|^2 \tag{9}$$

There are three possibilities when interpreting the result of the face recognition: Recognition: The input image can be assigned to a stored image in the database; Detection: The input picture was accepted as a image but could not be found in the data base;

Rejection: The input picture was not accepted as an image.

In order to classify the input picture we use appropriate threshold values.

$$\theta_{threshold} = \frac{1}{2}\max\left(\sqrt{||\Omega - \Omega_k||^2}\right) \tag{10}$$

The implemented system contains an expandable database for both the training images and the images and data of the registered persons.

4 Results

Myra system was tested several different ways. Face detection was tested on TV shows and films (TV Tuner), camcorder, analogue video recorder and webcam, mainly indoors. Face recognition was tested on webcam and stored and live camcorder video streams. The face detection delivers quite good results by appropriate head orientation 92% of the faces are found, but it deteriorates significantly by bad positions. The skin colour detection works (using 50% coverage) to 96% correctly, that means that it delivers remarkably reliably results, apart from extraordinary light conditions like back light or very dim lighting. The proportion of faulty positive detections for the skin tone detector depends on the background, with green grass and blue skies in the back it is 0%, with wooden, skin-color furniture or wallpaper it is obviously more. The hit rate of the AdaBoost method is 89% if the faces are frontal. The rate of erroneously positive detections varied in the analysed sample around 8% that can be reduced to 1% by means of skin color detection.

During the experimental examinations on the adequate images of the “Stirling” database, i.e. on faces looking straight into the camera, the face recognition achieved a recognition rate of above 70%. This rate increased for images on uniformly white background, where yet the head shape could be considered, to 85%. For inadequate images, e.g. for pictures of inappropriately illuminated faces the recognition rate could go down to below 50%. This is symptomatic of the holistic eigenfaces method that relies on pixel intensities and responds rather severely to changes of lighting. It is recommendable to add a method to the system that uses geometrical features, to make the face recognition more accurate. Using the “Gábor wavelets” for example should improve the system a lot, because they are utterly insensitive to light conditions The implementation of the Gábor wavelets method is planned in the following stage of our project. Combining both methods will enhance recognition quality but require more computing power. The goal is to make a face recognition system that decides upon prevalent circumstances which method it should prefer. In case that the pixel intensity of a sample differs strongly from that of the pictures in the database, the system should not attempt to use the intensity-based eigenfaces method but begin exclusively with the geometrical features method. Some figures about the speed of the face recognition module (using AMD 2700 Athlon, 500 MByte RAM and Windows

XP operating system): processing photos only in the face recognition module in silent mood (without displaying the results) leads to 18-19 recognition per second. Using the complete MYRA system to process video pictures with face detection and face recognition allows the processing of 10-11 frame per second.

Conclusions

Developing an application for face recognition that is reliable under changing circumstances is a big challenge. Myra system already works well under ideal conditions. Next targets are chiefly to make the system more flexible and quicker, to accommodate it to various life situations and scenes.

Acknowledgement

The research has been supported by the OM TDK grant under the terms of grant OM FPO 245540/2005.

Figure 4

The main (user) view of the system

Figure 5

Result of the face detection and the skin tone detection

References

[1] B. Bors, R. Horváth, M. Safranka, Z. Vámossy: CERBERUS Project, Human Verification: a Biometric Approach, in Proceedings of IEEE International Conference on Computational Cybernetics, Siófok, Hungary, August 29-31, 2003, CD issue

[2] M. M.-H. Yang, D. J. Kriegman, N. Ahuja: Detecting Faces in Images: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 24, No. 1, 2002, pp. 34-58

[3] W. Zhao, R. Chellappa, A. Rosenfeld, P. J. Phillips, Face Recognition: A Literature Survey, ACM Computing Surveys, 2003, pp. 399-458

[4] P. Viola, M. J. Jones: Rapid Object Detection Using a Boosted Cascade of Simple Features, in Proceedings of IEEE CVPR, Vol. 1. 2001, pp. 511-518

[5] Rein-Lien Hsu; Mohamed Abdel-Mottaleb; Anil K. Jain: Face Detection in Color Images, in: IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, May 2002, pp. 696-706

[6] Paul Viola; Michael Jones: Robust Real-time Object Detection, Cambridge Research Laboratory, 2001

[7] Michael Isard; Andrew Blake: Contour tracking by stochastic propagation of conditional density, in Proceedings of European Conference on Computer Vision, Vol. 1, Cambridge UK, 1996, pp. 343-356

[8] M. Kirby, L. Sirovich: Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990

[9] M. Turk, A. Pentland: Face Recognition Using Eigenfaces, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1991, Maui, Hawaii, pp. 586-590

[10] Jacobi Transformation of a Symmetric Matrix, Numerical Recipes in the Art of Scientific Computing, Cambridge University Press, 1988