Face Detection by Means of Skin Detection

Vitoantonio Bevilacqua[1,2], Giuseppe Filograno[1], and Giuseppe Mastronardi[1,2]

[1] Department of Electrical and Electronics, Polytechnic of Bari,
Via Orabona, 4 - 70125 Bari - Italy
bevilacqua@poliba.it
[2] e.B.I.S. s.r.l. (electronic Business in Security), Spin-Off of Polytechnic of Bari,
Str. Prov. per Casamassima Km. 3 – 70010 Valenzano (BA) - Italy

Abstract. In this article we present a novel approach to detect face in color images. Many researchers concentrated on this problem and the literature about this subject is extremely wide. We thought to decompose the overall problem into two intuitive sub-problems: the research of pixels have skin color in the original image and the analysis of pixels portions resulting, by means of a neural classifier. The achieved results show the robustness of presented approach.

1 Introduction

Face detection is the process has the goal to determine whether or not there are any faces in a digital image and, if present return the image location of each face. This activity is very simple for a man but is very hard for a machine because face detection isn't problem translatable in a mathematic language.

Face detection is a useful task because is a process propaedeutic to facial feature recognition, face recognition e facial expression recognition. All this activities, as a whole, aims to realize video-surveillance system, man-machine interaction and useful robotic application.

The most innovative application about face detection is the realization of integrated circuits that catch faces in a scene in order to get a photo in focus in a photo camera automatically.

In these last years many researchers are developed more than 150 approaches to solve face detection [1]. We can classify this problem in 4 principal methods:

- knowledge-based methods, in which the human knowledge of face is codified in any rules comprehensible to machine;
- template matching methods, that determine image region that have a face with template that describes them entirely or in a part;
- appearance-based method, in which there is an intelligent element that, with a series of examples, recognize a face in different condition;
- feature invariant approaches, that solve the problem finding a structural characteristic of a face (e.g. eye, skin, nose, etc.), in order to find an human face.

Knowledge-based methods are based on intuitive rules that derive by the human observation of a man. Face characteristics are related between them in fact all people have two eyes, a nose, a mouth placed to an opportune distance or that exist a symmetry property on a face. Yang e Wang used a hierarchic knowledge-based method [3].

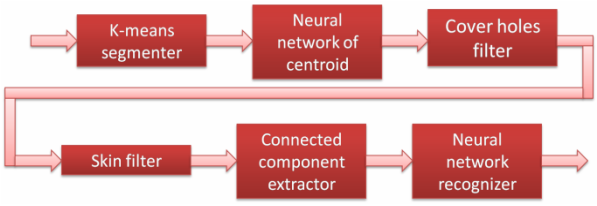
D.-S. Huang et al. (Eds.): ICIC 2008, LNAI 5227, pp. 1210–1220, 2008.

Their system consists of three rules levels. In the high level, all possible candidate faces are extracted with a window that scans the image. These candidates represent image portions that satisfy the rules of high level, and they are general description of a face by people. In low levels the rules represent the composition of structural components of a face, for example in the central part of face the pixels have a constant value (in gray scale), and there's a relation between central and contour pixels of a face. Beyond these rules the application performs some manipulations on the image. In fact at first level there's a resolution reduction of the input image, then the algorithm executed an histogram equalization of gray levels for potential faces and finally there is an edge extraction. The test on 60 images demonstrates that 50 of this are processed correctly.

Template matching methods find a face applying one or lots of templates that scan the input image. A first attempt of face detection whit this method reported by Sakai et al.[5]. They detect faces by means of two phases: the first aim to localize the regions of interest and the second confirms that these regions are faces with different sub-templates for eyes, nose, mouth and so on. The regions of interest searched transforming the original image in linear segments and performing the match with a template representing face contours. Finally, using each sub-template in each region, they calculated a correlation function to verify the face existence. Analyzing the algorithms realized with template matching, we can note that this technique is unlike for face detection because there is a strong dependence by form, scale and face pose. To delete this problem in these years invented multiscale, deformable and multiresolution sub-templates. Appearance-based methods are based on statistic techniques or on machines learning to find relevant characteristics of faces and non-faces. The best classifier proposed by Viola and Jones, built using AdaBoost learning algorithm [2]. Finally, feature invariant approaches are based on the idea that a man detect face and object in different poses and illumination conditions in a simple mode, so must exists any invariant characteristics on vary revelation conditions. In this method the application determine the facial characteristics first and the face then. Example of facial characteristic are eyes, eyebrows, nose, lips, but also texture (of face, hair and eyes) and skin color. Sirohey have proposed a face localization method to segment a face when there is an heterogeneous background [4]. In this work is used a Canny filter to edge extraction in the input image and a metric to delete any edge and group others. This processing is useful to have a face with only one edge. Finally the application fits an ellipse between head and background. With this method we can notice an accuracy of 80 % on test images. In the Section 2 we present our algorithm. We thought to decompose the overall problem of face detection into two intuitive sub-problems: the research of pixels have skin color in the original image and the analysis of pixel portions resulting, by means of a neural classifier. The solution of the first sub-problem is explained from the sub-section 2.1 to 2.4, while the second sub-problem is presented in the sub-section 2.6. The sub-section 2.5 is the bridge about the two sub-problems because aim to subdivide the just processed image into congruent portions to process called connected components. Finally, the Section 3 outlines the force and weak points of the proposed method.

2 Application Description

The application realized and described in this document, don't belong to anyone of four category presented but it's an hybrid method. In fact, it's an application in part feature invariant and in other part appearance based. It's feature invariant because the first portion of program aim to extract skin patches in a color image, but it's also appearance based because exists a neural face recognizer, that verify if a region hold a face. This application is developed in C++ language and it works whit 24 bit color digital images in any image format. All the functions on image processing manually realized, without any library, but the opening of the images performed about Qt libraries, useful to implement a GUI, too (for more information please visit the URL: http:// trolltech.com/products/qt/). The application can be schematized with this block diagram:



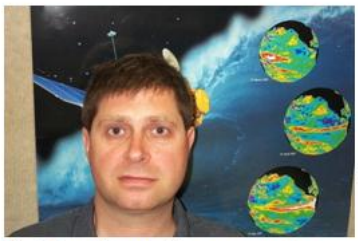


Fig. 1. Input image to K-means segmenter

Fig. 2.

Fig. 3.

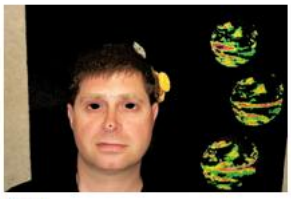


Fig. 4.

Starting by input image at this is made skin detection. Skin detection is the process of extraction of all pixels having skin color. We realized this activity segmenting the image first, and applying a model to describe skin color after. The application then extracts any connected components. For each component are derived some subcomponents, which will process to produce useful output for neural face recognizer. This last module will verify if a subcomponent is a face.

In next subparagraphs we will present theoretically and practically all modules of realized application.

2.1 K-Means Segmenter

K-means segmenter take care of image segmentation, applying a clustering algorithm called k-means. This algorithm starts by hypothesis that the number of cluster, in which will be divided the data set is k. In this case, the data set is the chrominance value of pixels image in CIE-L*a*b* color space. This color space is more useful than RGB because we noticed better output. In fact we performed a series of comparison tests between the two color spaces, that demonstrate how CIE-L*a*b* consents to extract optimally pixel having skin color. The application so operates a conversion from RGB to CIE-L*a*b* color space. In this last space each pixel is represented as combination of three components: L* the luminance component, a* and b* the chrominance components. In k-means segmenter L* component isn't used because it have only luminance informations, while we are interested to chrominance informations. This is another point in favour to CIE-L*a*b* color space because rather than execute k-means algorithm with three chrominance centroids components we used only two components, and this consents to saving useful time. If the segmenter have in input Figure 1, the output consists of three segmented images (Figure 2, 3, 4), of which only one will continue the next computations. Number three is just the value of k. This allow to segmenting in a good manner images that have an uniform background. For heterogeneous background (that are more frequently background) need make other processing, that we'll show in the next sections. K-means algorithm makes clustering based on distance concept of each element by cluster centroid to which it belongs. In various iteration of this segmenter three centroids move up to find a position in which the average of distance of each cluster element by its centroid is minimum. When the movement becomes null the algorithm is stopped and are created the three images (e.g. Figure 2, 3 and 4).

2.2 Neural Network of Centroids

The value of three centroids is the input of a multi-layer perceptron (MLP) neural network feed forward, whose objective is to detect what the three images have more probability to hold a face. This neural network is made up of an input layer of 6 neurons, an hidden layer of 9 neurons and an output layer of 3 neurons. Neural network of centroids have in input the value of three centroids of image and it produces in output the probability that image holds a face. This neural network trained using error back propagation algorithm with in input the centroids of 65 images containing faces.

The validation set formed of centroids of 28 images and the ANN reported always correct results. The values centroids for Figure 1 are:

121.370009	107.760515
124.026731	127.468500
135.179397	158.241178

And the neural network output is:

0.000020 0.024462 0.983285

The image that continues the computation is Figure 4 because it associated to highest neural network output.

2.3 Cover Holes Filter

After the neural network we realized a cover holes filter, that aims to clean entire image deleting the vacancy introduced by k-means segmenter. This filter implemented by means of a series of sliding windows of various dimensions proportional to image area. If contour pixels of window are all coloured and within there are black pixels, then this are re-established to original value. If the input of filter is Figure 4 then the effect is proposed in Figure 5.

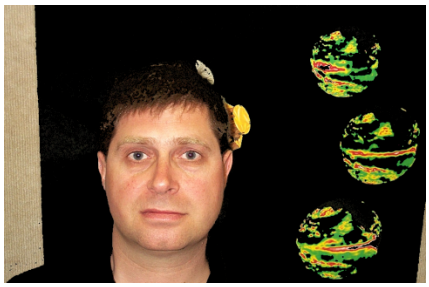


Fig. 5. Cover holes filter output when in input there is Figure 4

2.4 Skin Filter

Skin filter consists of implementation of elliptical boundary model for skin color. This model describes in a good manner the skin color distribution in all chrominance space by means of two parameters [6]. The inventors of this model verified that the skin color distribution in opportune chrominance spaces (r-g normalized and CIE-xy) very well approximated by an ellipse. We realized this model implementing a three-dimensional histogram in r-g normalized space, obtained counting chrominance values of pixels in 44 example skin patches (Figure 6).

In Figure 7 there is the 3D histogram obtained and in Figure 8 there is the same histogram seen in an up view. In this second view we can see that skin color pixel

Fig. 6. Skin patches used for implementation of three-dimensional histogram

thicken in an elliptical zone. If we indicate with X the chrominance value of pixel, the ellipse can be described by this formula:

$$\Psi(X) = (X - \Psi)^T C^{-1} (X - \Psi) \tag{1}$$

Where Ψ and C are the average values of chrominance vectors and covariance matrix respectively. The values of Ψ and C are this:

$$\Psi = \begin{bmatrix} 0.495271 \\ 0.25982 \end{bmatrix} \qquad C = \begin{bmatrix} 0.0036 & -0.0015 \\ -0.0015 & 0.0013 \end{bmatrix}$$

We tested the skin filter with a series of images discovering that the best threshold value that consents to reject pixels whose color is different by skin color is θ=5. If Φ(X) is less than θ then pixel is skin, or else is blackened. Equality Φ(X)=θ identifies an ellipse in r-g normalized color space, when Ψ and C are the center and ellipse principal axes. In Figure 9 is possible to analize the effect of skin filter when the input is Figure 5. The elliptical boudary model isn't the only study about skin detection, similar works are developed by Terrillon et al.[7] have used single gaussian skin model and by Yang and Ahuja[8] that have implemented the mixture of gaussian skin model. We've preferred elliptical boundary model because at the state of art it's the

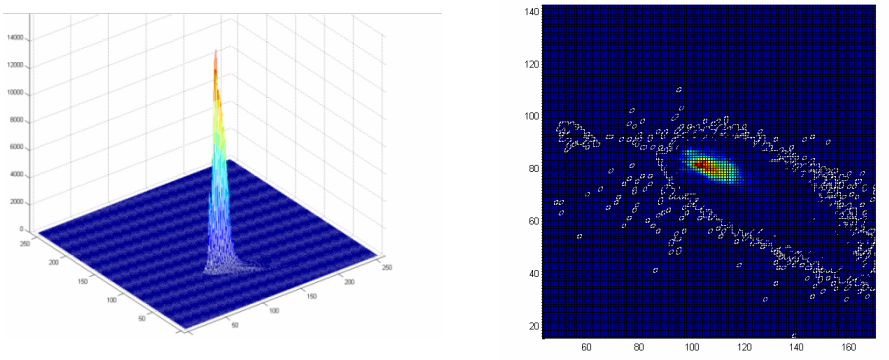


Fig. 7. Three-dimensional histogram of skin distribution and its up view

Fig. 8. Skin filter output

best model for skin detection. However for a complete presentation of skin color model you can refer to Yang, Kriegman et al.[9].

2.5 Connected Component Extractor

After skin extraction, we can search connected component in image, but, to make this, the image must be transformed in binary. So color pixels are set to 1, while black pixels to 0. A connected component is an image portion composed of a pixels set that verify 4-connection property [10]. A pixel is 4-connected if at least one of its 4 near pixels (placed in right, left, up and down) are set to 1. Really all components didn't considerate, because also lone pixel acts as connected component. So we have introduced a check on components area, which must be above the 0.05 % of image area so that the component can proceed in next computations. We've chosen to delete components that have this area value because it consents to delete the noise visible in Figure 8. At this point the application sequentiality stops for introduction of some thread. In fact for each connected component we created a thread that takes care of: extract some sub-components in square for, magnify or reduce them to make them multiple of a window 20x20 and finally equalize gray levels histogram to amplify some shades. Moreover for each subc-omponent we calculated 404 means with two sliding window with different form, so that it occupies 404 position. These positions are visible in Figure 9 and for the second face we can note that the average is calculated on 4 rows blocks. This is made because the 4 averages contain the fact that the eyes and mouth are more dark than central part of a face.

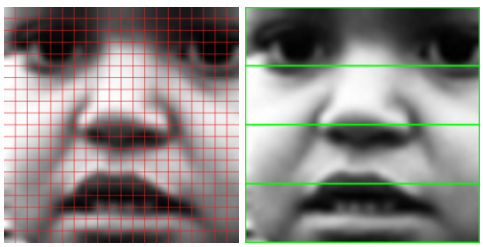


Fig. 9. Positions of sliding window when application calculate 404 means

2.6 Neural Face Recognizer

The neural face recognizer is a MLP neural network able to detect if a subcomponent is a face. Its architecture is made of an input layer of 404 neurons, a hidden layer of 200 neurons and an output layer of a neuron, and all neurons have as activation function an hyperbolic tangent sigmoid function. The network accepts in input 404 means calculated for each subcomponent and it produces in output the probability that this subcomponent is a face.

This network is trained with face and non-face examples (see Figure 10), using error back propagation algorithm. In fact we have used 80 different images face at various dimension for a total of 719 images, while for non-faces we have employed 441 examples contained skin portions of leg, hand, neck and so on. The calculated means are normalized in interval [-1,1] to represent congruent input for neural network. If the output of recognizer overcomes the threshold value 0.8 than the thread will apply in original image a colored square. This threshold value allows to reach trade-off between false positives and false negatives.

Fig. 10. Examples of face and non-faces used to learn Neural Face Recognizer

3 Test and Conclusion

The application is tested on a set of 50 images searched casually on the web. In this set 10 images have only one face, 7 no face and 33 are group photos. The neural network of centroid work good in 96 % of cases, so only 2 images on 50 aren't analyzed correctly by next modules. Excluding this two images the test results are:

- detection rate of 64.04 % and detection rate of 90 % for images containing one face (this means that 9 face on 10 are detected);
- 36 % of false negatives and 35, 96 % of false positive (i.e. 67 false negatives on 48 test image).

Compared to literature we can derive that application detected faces in a good manner in optimal condition. For optimal condition we intend uniform background and/or face at intermediate distance by camera. Doing a comparison with developed application by Rowley et al.[11], that used MLP neural network, he obtain a detection rate of 90 %, having used 1050 face and 8000 non-faces for training. In our application we have 66 % of detection rate using 80 faces and 441 non-faces. Future developments will provide an improvement of detection rate and a reduction of false positive. Initially this application didn't support threads and execution was slow. Execution time is a crucial factor in video surveillance application or in robotics. In our case this time depends on image to process and on number and extensions of connected components. With threads

introductions execution time is lowered, saving from few seconds, for images whose execution time is 25-30 seconds, to 16 seconds, for images more complex whose execution time is 86-120 sec. We have chosen ANN as classifier just to reach a trade-off between detection rate and this important parameter. We must spend few words about skin detection technique used. The fact of skin pixel are detected with first four application blocks allow us to realize the skin filter using a number really low of skin patches, e.g. 44 pictures versus 4675 images of Lee[6]. In fact the presence of k-means segmenter consents to reject most of background pixel in order to allow a refining work for skin filter. Next table summarizes comparisons relating to Rowley algorithm [11] and our method.

Figure	**Dimensions**	**Execution time (s)**	**Detection Rate (%)**		**False Positives**		**False Negatives**	
			Rowley	Bevilacqua	Rowley	Bevilacqua	Rowley	Bevilacqua
11	800x600	19	100	100	0	0	0	0
12	1024x768	39	100	100	0	1	0	0
13	1024x890	77	55.56	66.67	0	3	4	3
14	1024x769	89	86.36	68.18	0	6	3	6
15	800x600	292	0	29.03	0	0	31	22

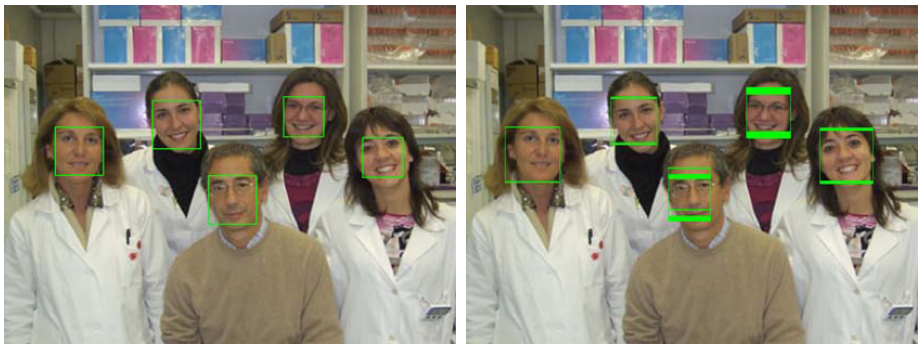


Fig. 11. Comparison between Rowley (left picture) and our algorithm (right picture)

Fig. 12. Comparison between Rowley and our algorithm

Fig. 13. Comparison between Rowley and our algorithm

Fig. 14. Comparison between Rowley and our algorithm

Fig. 15. Comparison between Rowley and our algorithm for very small faces

References

1. Yang, M., Kriegman, D., Ahuja: Detecting Faces in Images: A Survey. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(1) (2002)
2. Viola, P., Jones, M.: Robust Real-Time Face Detection. International Journal of Computer Vision (2004)
3. Yang, G., Huang, T.S.: Human Face Detection in Complex Background. Pattern recognition 27(1), 53–63 (1994)
4. Sirohey, S.A.: Human Face Segmentation and Identification. Technical Report, University of Maryland (1993)
5. Sakai, T., Nagao, M., Fujibayashi, S.: Line Extraction and Pattern Detection in a Photograph. Pattern Recognition 1, 233–248 (1969)
6. Lee, J., Yoo, S.: An Elliptical Boundary Model for Skin Color Detection. In: International Conference on image science (2002)
7. Terrillon, J., Akamatsu, S.: Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images (1999)
8. Yang, M., Lu, W., Ahuja, N.: Gaussian Mixture Model for Human Skin Color and its Application in Image and Video Database. In: Conference on Storage and Retrieval for Image and Video Database (1999)
9. Yang, M., Kriegman, D., Ahuja, N.: Detecting Faces in Images: A Survey. IEEE Transaction on pattern analysis and machine intelligence 24(1) (2002)
10. Gonzalez, R., Woods, R.: Digital Image Processing. Prentice-Hall, Englewood Cliffs
11. Rowley, H., Baluja, S., Kanade, T.: Neural Network-Based Face Detection. IEEE Transaction on pattern analysis and machine intelligence (1998)