A Fast and Accurate Faces Localization using Gradient Method

Georgi Kukharev, Pawel Masicz, Piotr Masicz
Technical University of Szczecin
Faculty of Computer Science and Information Technology
Zolnierska 49, 71-210 Szczecin, Poland
gkukharev@wi.ps.pl; gmasicz@wi.ps.pl; pmasicz@wi.ps.pl

ABSTRACT

Detecting faces in images with complex background is a difficult task. To solve this problem we decided to use modified gradient method with oval object detection. This method is very fast and our improvements give better accuracy than it was in original gradient methods. Our system was tested using pictures delivered from digital cameras, camcorders and television. On those various sources, differences in head size, lighting and background are considerable. During experiments we achieved very good results – 93-99% of correct face localization. We also implemented our method in C++ - it works well and fast.

Keywords

Face localization, gradient method

1. INTRODUCTION

To localize a face in an image means to find its position and its size or scale. It is very important to do it correctly if we try to build complex system for face processing such as visitor identification or other security system [Kuk01a, Lan95a], because all next calculation strongly depends on it. Face localization and detection in images is not easy because face are non rigid and have a high degree of variability in size, shape, color, orientation and pose [McK98a, Cra92a, Kje96a, Fas98a]. Diversity of background does not help in this process also. Other problem which is necessary to solve is "face/nonface" classification. During last ten years there were published many approaches solving face detection and localization problems. Most of them depends on Principal Components Analysis (PCA), neural network, machine learning, Hough transform, movement extraction, projections and color analysis [Cha98a, Osu97a, Row98a, Row99a, Sun98a, Sun98b, Ter98a, Yan92a, Yan02a, Yow97a]. All of them have weak points, for example solutions which based at neural network requires lots of testing images (faces and nonfaces; results of working strongly depends on training sets) and in most cases are built to detect faces in frontal position in grayscale image. Color segmentation does not work when there are lot of skin-like color areas near face. In most cases, group of methods which based at holistic representation works better than other, when faces on images are small and when image quality is rather poor. Other methods, which based on geometrical features, have better results for rotated faces. When system uses face's color information there is a problem with poor resistance on various lighting condition and specific background. In section 2, we introduce the original Gradient method. Section 3 describes our proposition to improve this method. In section 4 and 5, results and summary are presented.

WSCG SHORT Communication papers proceedings
WSCG'2004, February 2-6, 2003, Plzen, Czech Republic.

2. GRADIENT METHOD WITH OVAL OBJECT DETECTION

All of known gradient methods have one shared feature which depends on examination of local luminance value changing in images. In most cases, to compute the matrix of gradients, grayscale images are used because color information is not necessary. Data which are achieved from this computation could be applied to work in many problems, for example to edges and shapes detection – including oval object [Kuk03a].

First step in original gradient method with oval object detection is rescaling source image to size 120x90. Next it is applied disproportionate rescaling in way that face shape become an oval from elliptic. This step strongly decreases amount of calculations.

Experiments proved that width of face is close to 75% of its height which is showed on "Figure 7". Next, we apply normalization step (Fig. 7b) which standardize characteristic of pixels taken from source images. To do this there are applied histogram equalization and filters. Next, there are computed two matrixes – Gx i Gy (Fig. 1b, 1c) which contain information about local luminance changeability in two planes – vertical and horizontal:

$$Gx_{y,x} = \frac{1}{2}\left(P_{y,x+1} - P_{y,x-1}\right), \quad (1)$$

$$Gy_{y,x} = \frac{1}{2}\left(P_{y+1,x} - P_{y-1,x}\right), \quad (2)$$

where:
$P_{y,x}$ - value of color in point [y,x],
$Gx_{y,x}$ - value of horizontal gradient in point $P_{y,x}$,
$Gy_{y,x}$ - value of vertical gradient in point $P_{y,x}$.

For each pixel of image length of gradient's vector is computed using simply equation:

$$G_{y,x} = \sqrt{(Gx_{y,x})^2 + (Gy_{y,x})^2}, \quad (3)$$

where:
$G_{y,x}$ - value of gradient in point $P_{y,x}$,

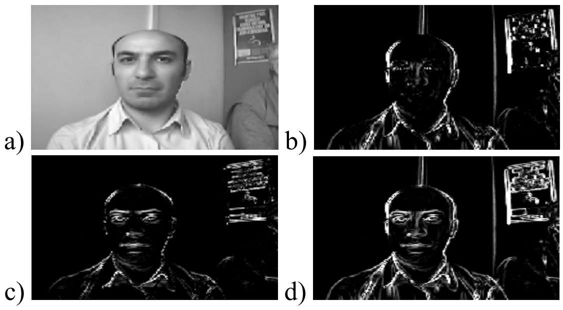

Figure 1. Source image (a) and matrixes of gradient – Gx (b) and Gy (c) and their sum (d)

If length of vector G is known, all points where gradients have too small value are automatically eliminated from next calculation. This step allows us to decrease amount of calculations and improve accuracy of method because we assumed that points where value of gradient are too small does not contain information about human face shape. In generally, the procedure, which allows us to find centre of human face in source image, depends on three parameters: value of horizontal gradient Gx, value of vertical gradient Gy and radius R. All points in image are computed (except points which are eliminated in previous step) in way that is presented below (Fig. 3).
The "hit" points we calculate using equations which are easy to explain basis of figure 3:

$$R^2 = (Gx'_{y,x})^2 + (Gy'_{y,x})^2, \quad (4)$$

$$G_{y,x}{}^2 = (Gx_{y,x})^2 + (Gy_{y,x})^2, \quad (5)$$

$$\frac{Gx_{y,x}}{G_{y,x}} = \frac{Gx'_{y,x}}{R}, \quad \frac{Gy_{y,x}}{G_{y,x}} = \frac{Gy'_{y,x}}{R}. \quad (6)$$

After transformations we received:

$$Gx'_{y,x} = \frac{R \cdot Gx_{y,x}}{\sqrt{Gx_{y,x}{}^2 + Gy_{y,x}{}^2}}, \quad (7)$$

$$Gy'_{y,x} = \frac{R \cdot Gy_{y,x}}{\sqrt{Gx_{y,x}{}^2 + Gy_{y,x}{}^2}}, \quad (8)$$

The centre of face coordinates is calculated using:

$$S_{y,x} = (x + Gx'_{y,x}, y + Gy'_{y,x}). \quad (9)$$

Aforementioned computations are applied for every pixel of source image. To store achieved results it is created a special matrix called "matrix of hits" (all their values at start are 0). Each S coordinates, which are returned from algorithms described above, increase values of pixel in "matrix of hits" using dependence visible in "Figure 2" (coordinates for point contained value +3 is identical with coordinates for point S). If value of searching radius is equal to face's radius, many of computed points "hit" point near centre of this face. "Figure 4" helps to understand this idea.

	+1	+2	+1	
	+2	+3	+2	
	+1	+2	+1	

Figure 2. The way to fill up "matrix of hits"

Symbols used in "Figure 3":
S - centre of face (probable),
R - radius of face (expected),

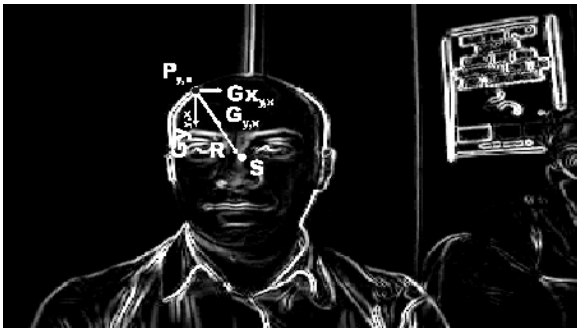

Figure 3. Illustration of way to achieve Gx'$_{y,x}$ i Gy'$_{y,x}$ basis of value of gradients and radius R

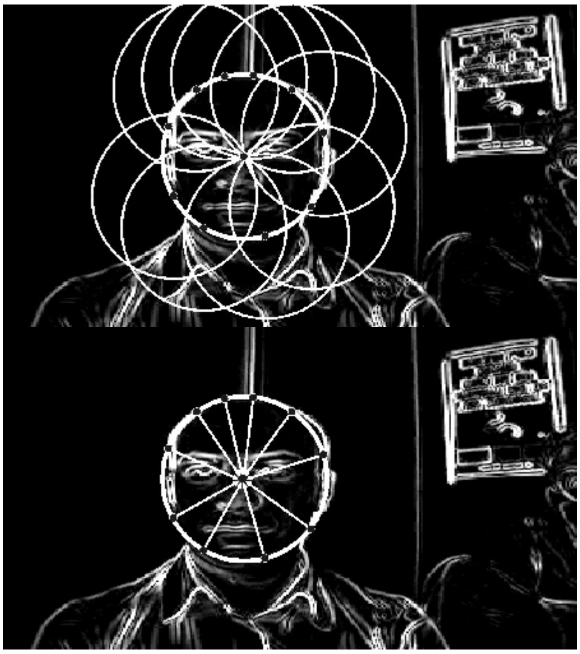

Figure 4. Illustration of how searching radius "hits" the centre of face when it is equal to face radius

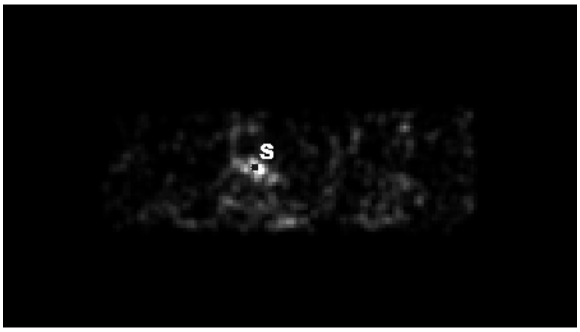

Figure 5. "Matrix of hits" when searching radius is equal to face radius

All previous calculations give back a matrix which contains information about hits for radius R. In "Figure 6" it is clearly visible that in neighborhood of centre of face is a largest cluster of pixels contained high value in opposition to rest. There is also the pixel with highest value. In high probability it is the centre of finding face. It is necessary to add information that in original gradient method face must have identical or very close radius to searching radius R because only then received "matrix of hits" and results could be correct. To solve this problem, pyramids of images are used (Fig. 7). This solution is very time-consuming, but it allows us to execute original algorithms with one, constant radius R. Source image is rescaled to different resolutions and then face have different size. The assumption, that in one of this rescaled images radius of face is equal to searching radius R, is logically.

Figure 6. Pyramids of images with results of computation for constant R

Illustration of gradient method with oval object detection, which is described in this chapter, is showed below (Fig. 7).

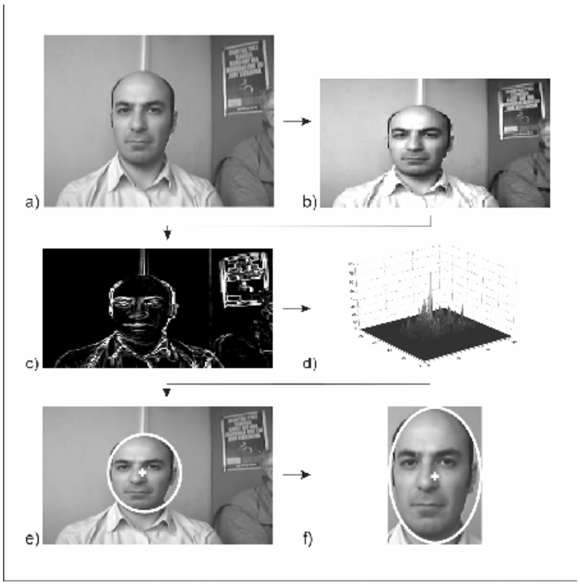

Figure 7. Scheme of work for gradient method with oval object detection: source image (a), after normalization and disproportionate rescaling (b), matrix of gradient (c), "matrix of hits" (d), results of findings (e), extracted face (f)

There are pictures showed below (Fig. 8) which illustrate correct results of gradient method with oval object detection. We can see that gradient method is very resistant in various lighting condition. It is also worth noticing that various background does not affect on results. Unfortunately, when radius of face does not equal to searching radius R results achieved from algorithm are wrong (Fig. 9b). Other objects which have oval shape could also generate wrong results (Fig. 9a).

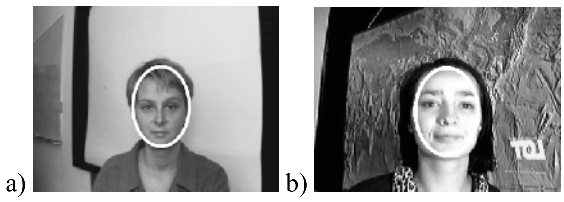

Figure 8. Examples of correct working

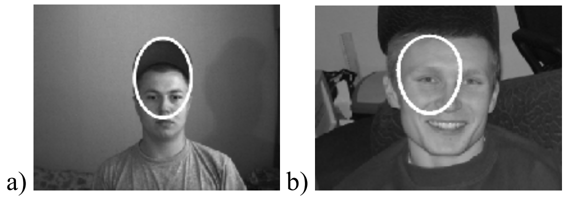

Figure 9. Examples of wrong working

3. EXTENDED GRADIENT METHOD WITH OVAL OBJECT DETECTION

As we wrote in abstract, gradient method described above has one undeniable advantage – speed. This advantage makes it very interesting to implement in real-time systems. But it has also few weak points: only grayscale images are possible to compute (color information is omit), constant radius R (face must have right, known size), accuracy (not very high). Our improvements minimize those weak points.

First step is proportionate rescaling source image (320x240) to resolution 60x45 and then disproportionate rescaling to 80x45 (Fig. 11a). The ratio between 60 and 80 is equal to 75% which is exactly the same as ratio between head's height and width. As we can see on "Figure 10a" face shape after rescaling is very close to oval in opposition to elliptical which was before. This solution gives one additional advantage – improves speed, because amount of calculation is much smaller when we trying to find oval shapes. Other improvement is to use I2 component from modified I1I2I3 color space:

$$I2 = R - G \tag{10}$$

where:

R, G - components from RGB color space.

When we use component I2, usable color information is not lost. We use one additional normalize step which depend on cutting and histogram equalizing (Fig. 10b). "Figure 10b" shows us, that using component I2 in opposition to grayscale makes possible to narrow down area where we search a face. It reduces amount of calculation and improves accuracy because objects, which have not color similar to typical face, simply disappeared.

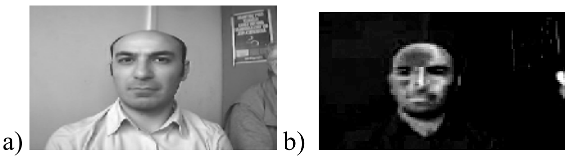

Figure 10. Rescaled source image in RGB color space (a), in I2 color space with normalize step (b)

Image prepared in way described above is passed to module which calculates gradient matrixes (Fig. 11a,b). To improve accuracy of gradient method we calculate two additional gradient matrixes – G45 and G135 (Fig. 11c,d):

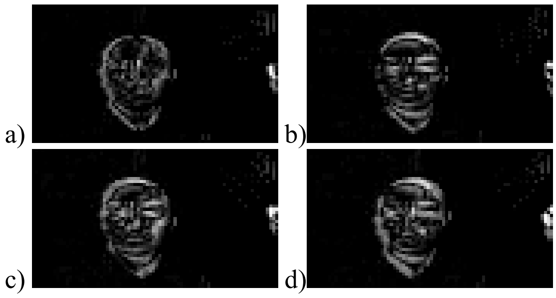

Figure 11. Illustration of matrixes of gradient Gx (a), Gy (b), G45 (c) and G135 (d)

We decided to use two additional matrixes (G45 and G135) because original gradient method process data only in two directions – vertical and horizontal and omit information in diagonal.

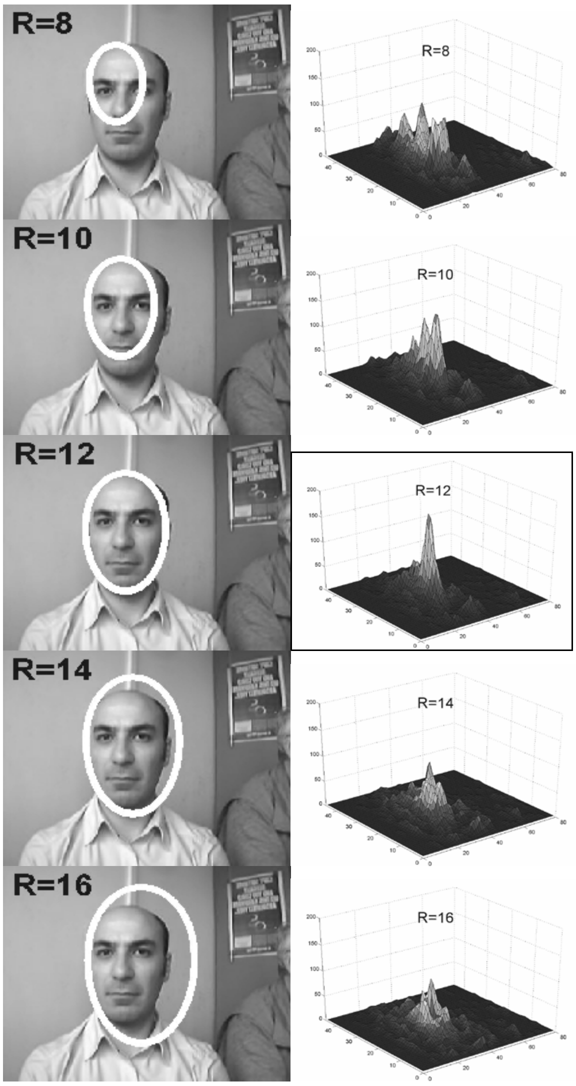

Figure 12. Source image and "matrixes of hits" for radius R which values are contained in 8-16 pixels (step is equal to 2, the best result is also shown by black frame)

In our algorithm this two additional matrixes are computed in the same way that Gx and Gy. Achieved results are wrote to the same “matrix of hits”. We decided to modify original algorithm which depends on pyramids of images because image rescaling is very time-consuming. Our new approach uses variable searching radius R in opposition to constant in original method. The R radius is contained in 15-50% compartment of rescaled image’s height (results of algorithm’s work are visible in "Figure 13"). We also decided to use one collective “matrix of hits” which includes maximums from all “matrixes of hits” (because for every radius R, new “matrix of hits” is computed). This approach makes possible to extract many faces from image.

Our algorithm is very fast because number of loops is very low especially, when source image is rescaled to resolution 80x45. We assume that face must be entirely in picture so we do not search it near border of image. "Figure 13" help to understand this idea.

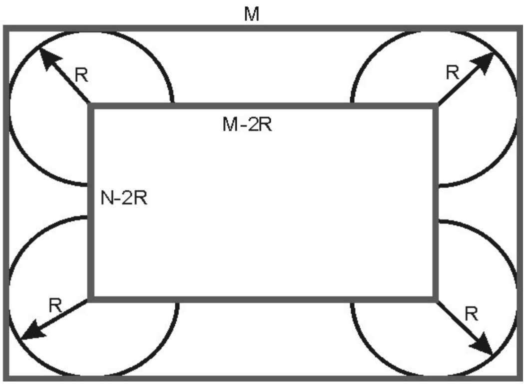

Figure 13. Illustration of idea where algorithm tries to find a face.

$$Q = 2 * \sum_{M/6}^{M/2} (M - 2 * R) * (N - 2 * R) \quad (11)$$

where:

MxN – resolution of source image,

R – searching radius (15-50% of N).

- if M=80, N=45 then Q=30000 (loops)
- if M=160, N=90 then Q=260000 (loops)

4. RESEARCH RESULTS

Most of databases, which allow to examine accuracy of face detection algorithm, are designed rather for image recognition systems, than for detection. Those databases often contain grayscale images, where faces are in centre and where background is not very cluttered. Even special databases, which were created to solve detection problem, have only grayscale images (for example Carnegie Mellon University (CMU), BioID Database [6] or FERET Database). That is why we created our own databases. First of them contains 179 pictures delivered from digital cameras, web-cams and Internet (Fig. 14). Most of those pictures have 320x240 resolutions in 24-bits color depth (JPG and BMP). Second one contains image sequence (1693 pictures) delivered from TV video stream (Fig. 15). Parameters: resolution – 320x240, color depth – 24-bits.

Figure 14. Few results of face detection in image delivered from digital cameras, web-cams and Internet (database 1)

To achieve high diversity of data we choose images in which faces have various size, color and pose. Lighting conditions are also different. Among examples in both of our databases (Fig. 14), there are faces located in background which has very similar color to face’s color. In that case systems depend of color segmentation have very hard problem to achieve correct results. We tested our algorithms to locate multiple faces (Fig. 16) and results are very good. Faces could encounter themselves and could be covered (no more than 50%). We successfully located even 10 faces on single image (resolution - 320x240). Our algorithms have one special advantage – locating multiple faces does not enlarge time-consuming because for each additional face it is only 1% more of overall computing time. That is because at the end of our algorithm we make one collective “matrix of hits” which contain, for each point, all maximums from all “matrixes of hits”. If we locate first maximum in this table, we can search next maximum, which show as next face candidate.

Figure 15. Few results of face detection in image delivered from TV video stream (database 2)

Figure 16. Example of multiple face detection

There are results of our research in "Table 2 and 3". We have studied effectiveness of our system depending on size of gradient matrix. We also tested it when two additional matrixes (for angle 45 and 135) is used and not used. As we can see, methods, which are proposed by us, are very useful in practice. We also proved that two additional matrixes (for angle 45 and 135) improve accuracy of gradient method, especially when source data are from poor quality video stream. In that case we noticed 10-15% improvement of accuracy. During our experiments we also noticed that smaller size of matrix of gradient is better to use than larger. That is because source image rescaling is some kind of normalization step (noise data from picture are eliminated). Fortunately, smaller size of matrix of gradient improves speed of our method.

Matrix of gradient size	Digital camera	TV video stream	Det. Time
60x45	168/179 93,8%	1509/1693 89,1%	10 +/- 5 ms
100x75	167/179 93,3%	1298/1693 76,7%	34 +/- 11 ms
200x150	165/179 92,2%	1128/1693 66,6%	252 +/- 52 ms
320x240	158/179 88,3%	1011/1693 59,7%	1,01 +/- 0,42s

Table 2. Illustration of effectiveness and detection time of our method depending on size of matrix of gradient (without 45/135 gradient matrixes)

Matrix of gradient size	Digital camera	TV video stream	Det. Time
60x45	174/179 97,2%	1596/1693 94,2%	13 +/- 5 ms
100x75	172/179 96,1%	1518/1693 89,6%	53 +/- 12 ms
200x150	188/179 93,8%	1351/1693 79,8%	308 +/- 59 ms
320x240	185/179 92,2%	1164/1693 68,7%	1,23 +/- 0,53s

Table 3. Illustration of effectiveness and detection time of our method depending on size of matrix of gradient (with 45/135 gradient matrixes)

5. CONCLUSION

During experiments we tested gradient method on various set of pictures. We proved that our method has possibility to locate even 10 faces in single image and works well in unfavorable condition like changing lighting condition, various background, even when face is partially covered. Especially this last advantage makes our method very special distinct from other detection method. We see possibility to next improvements, for example using other color space (HSV, YCrCb) for preprocessing and make from them better source to gradient method. It is also necessary to implement effective „face/nonface" algorithms with assumption that it must be fast. In near future we will focus on problem – to specify how many faces are in image.

6. REFERENCES

[Cha98a] Chai, D. and Ngan, K. N. Locating facial region of a head-and-shoulder color image. International Conference of Face and Gesture Recognition, pp. 124-129, 1998.

[Cra92a] Craw, I., Tock, D., Bennett, A. Finding Faces Features. Proc. Second European Conf. Computer Vision, pp. 92-96, 1992.

[Fas98a] Fasel. B. Fast multi-scale face detection. IDIAP-COM 4, 1998.

[Kje96a] Kjedsen, R. and Kender, J. Finding Skin in Color Images. Proc. Second Int'l Conf. Automatic Face and gesture Recognition, pp. 312-317, 1996.

[Kuk01a] Kukharev, G. and Tujaka, A. Pattern recognition methods for visitor identification and access control. Pattern Recognition and Information Processing, pp. 19-34, 2001.

[Kuk03a] Kukharev, G., Masicz, Pa., Masicz, Pi. Real-time face detection using modified gradient method. Pattern Recognition and Information Processing, vol. 2, pp. 195-200, 2003.

[Lan95a] Lanitis, A., Taylor, C. J., Cootes, T. F. An automatic Face Identification System Using Flexible Appearance Models. Image and Vision Computing, vol. 13, no. 5, pp. 393-401, 1995.

[McK98a] McKenna, S., Gong, S., Raja, Y. Modelling Facial Colour and Identity with Gaussian Mixtures. Pattern Recognition, vol. 31, no. 12, pp. 1883-1892, 1998.

[Osu97a] Osuna, E., Freund, R., Girosi, F. Training Support Vector Machines: An Application to Face detection. Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 130-136, 1997.

[Row99a] Rowley, H. A. Neural network-based face detection. PHD thesis, Carnegie Mellon University, Pitsburgh, 1999.

[Row98a] Rowley, H., Baluja, S., Kanade, T. Neural Network-Based Face Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, 1998.

[Sun98a] Sun, Q. B., Huang, W. M., Wu, J. K. Face detection based on color and local symetry information. International Conference of Face and Gesture Recognition, pp. 130-135, 1998.

[Sun98b] Sung, K.-K., Poggio, T. Example-Based Learning for View-Based Human Face Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 39-51, 1998.

[Ter98a] Terrillon, J. Ch., David, M., Akamatsu, S. Automatic detection of human faces in natural scene images by use of a skin color model and of imwariant moments. International Conference of Face and Gesture Recognition, pp. 112-117, 1998.

[Yan92a] Yang, G. and Huang, T. S. Human Face Detection In Complex Background. Pattern Recognition, vol. 27, no. 1, pp. 55-63, 1994.

[Yan02a] Yang, M.-H., Kriegman, D. J., Ahuja, N. Detecting Faces in Images: A Survey. IEEE Transitions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 1, pp. 34-56, 2002.

[Yow97a] Yow, K. C. and Cipolla, R. Feature-Based Human Face Detection. Image and Vision Computing, vol. 15, no. 9, pp. 713-735, 1997.