Brazilian Journal of Probability and Statistics
2009, Vol. 23, No. 2, 179–195
DOI: 10.1214/08-BJPS021

Combining information in a Bayesian network for face detection

Jorge Cândido and Maurício Marengoni

Universidade Presbiteriana Mackenzie

Abstract. Face detection has been an important research topic over the last 20 years. It is commonly used as a first step in face recognition and several techniques were applied in face detection, going from geometrical methods such as model matching to connectionist methods such as neural networks. This work presents a face detection system that uses a Bayesian network to combine information from different computational cheap visual operators, and is part of an ongoing project that uses a webcam to perform a reliable and accurate eye tracking. The face detection is the first step in this project. The aim in this work is to show that combining simple features in a Bayesian network helps improving the performance in a face detector system, increasing the detection rate and speeding up the face detection process.

1 Introduction

One of the main tasks in computer vision is object detection. Object detection is the first step in most vision tasks such as pattern recognition, tracking, or object classification. Object detection is a challenging step because, in general, there are no constraints on how the object shows up in an image. There are differences related to illumination, type of sensor used, visualization, and color, among others (Scheneiderman and Kanade (2000)). The detection of human faces is important due to its application in surveillance systems, human computer interaction (HCI), and biometrics systems (Yang, Kriegman and Ahuja (2002)). Despite the fact that some evolution has happened in the face detection area, there is no consensus about what method is the best for all situations in the real world. According to Yang, Kriegman and Ahuja (2002), the lack of a unique set of test images or benchmarks and a systematic performance evaluation makes it hard to evaluate algorithms and select the best one for each domain.

The work developed for face detection can be divided into two main groups: knowledge-based methods and appearance-based methods (Kadoury and Levine (2007)). The work presented here uses the knowledge-based method. The motivation for this work is related to human computer interaction and it is part of an ongoing optical mouse project for disabled computer users. The optical mouse concept designed here allows users with certain disabilities (e.g., Parkinson's disease) to operate or navigate in the Internet using the eyes to move the mouse cursor

Key words and phrases. Face detection, computer vision, Bayesian networks.
Received April 2008; accepted December 2008.

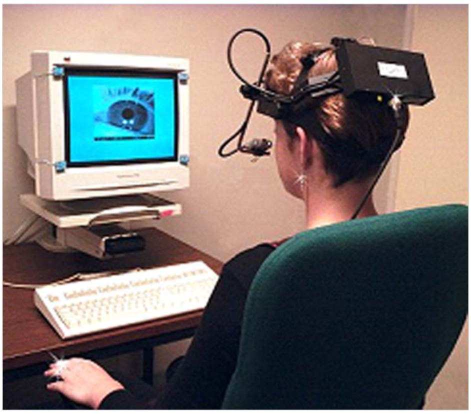

Figure 1 *Optical mouse with the main system fixed on the users head.*

and click on certain positions. There are a few systems that have part of these capabilities, but, in general, these systems are expensive and either they do not have the required accuracy or they impose movement limitations which restrict their usage for long periods of time; an example of this type of system is presented in Figure 1.

The system presented in Figure 1 is similar to the EyeLink II by SR Research (www.sr-research.com) which weights about 0.5 kg. Notice that it is probably not very comfortable to stay in front of a computer wearing this headset for a long period of time. In this project a regular webcam is used as an input device, to detect human faces in front of a computer screen. The image stream will be acquired in real time by the webcam. Once the face is detected, it will be tracked and, using a geometrical face model, the eyes expected position will be determined, tracked and, finally, the gaze estimation will be computed. Once the eyes are detected, a simple calibration process should provide enough accuracy for the optical mouse. The system will be composed by a face detector capable of detect and track a face, passing the eyes location to the next layer, an eye tracker whose job is to follow the eyes movement and compute the gaze estimation, in order to command the cursor on the screen. The schematic idea of the system is presented in Figure 2.

The use of multiple features to perform face detection is not new (see Masip, Bressan and Vitria (2005); Chai and Ngan (1998); Chen, Wu and Yachida (1995); Wu et al. (1996)); the use of Bayesian networks for face detection is also not knew (see Pham, Worring and Smeulders (2002); Sebe et al. (2005)). However, these two systems use the appearance-based method and learn the network from data. The difference in this work is related to the approach (knowledge-based) and the fact that the network is not learned. The Bayesian network is used to combine information from computationally simple, individual results in a probabilistic framework structure. Another main difference in this work is that the features are computed using different methods, which is not common—most systems try to find a face using a single method. It is shown that the correct selection of operator (feature

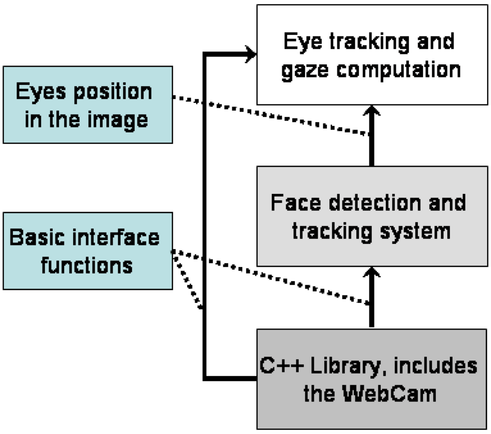
Eye tracking and gaze computation
Eyes position in the image
Face detection and tracking system
Basic interface functions
C++ Library, includes the WebCam

Figure 2 *The steps in the optical mouse project and their relationships.*

detector) and the Bayesian network's structure, increases the detection rate, decrease the number of false-positives, and the whole process is faster. The system was tested using two different datasets, one created specially for this project and the FRI CVL Face Dataset (http://www.lrv.fri.uni-lj.si/~peterp/FaceDB.zip), with similar performance.

2 Related work

In one of the most recent surveys on face detection (Yang, Kriegman and Ahuja (2002)), the systems which show the best detection rate were in the 90% to 95% range. More recent work in face detection show diferent techniques and/or applications but small increase in the detection rate. Kadoury and Levine (2007) system was designed to work with single images in grayscale. It uses locally linear embedding (LLE) for dimensionality reduction; the detection rate obtained was about the same as the others. Mamalet, Roux and Garcia (2007) works with real-time video streams and uses the convolutional face finder algorithm; its detection rate was about 90%. Masip, Bressan and Vitria (2005) presents an approach based on feature detection for real-time face detection and classification; the system works with a Bayesian classifier and presents a detection rate of 94%. Huang and Trivedi (2008) present a multiple task system to work with video streams; this system detects faces using skin colour and shape with an overall detection rate of 99%.

The classical definition of face detection (Yang, Kriegman and Ahuja (2002)) is: given any image is there a face in the image? If so, where in the image is the position of the face? This task is distinct from face recognition, where the idea is to match a face against a face database and return the data related to the matching face in the database. Figure 3 shows the idea of face detection, the input image has several faces and the output image marks all the faces in the image.

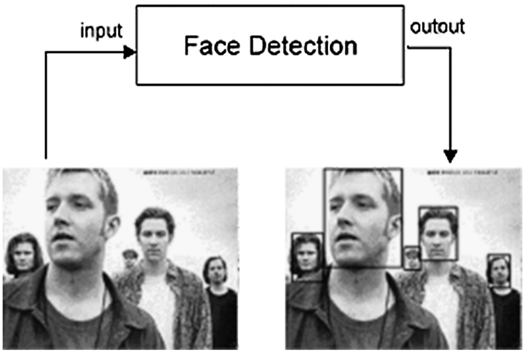

Figure 3 *The face detection process.*

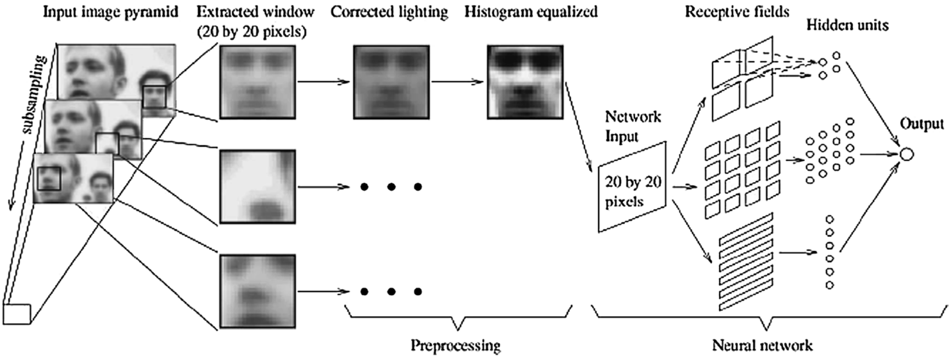

Figure 4 *The neural network scheme used in Rowley, Bajula and Kanade* (1996) *and Rowley, Bajula and Kanade* (1998) *for face detection.*

The problem of finding faces in digital images has attracted a lot of attention in the computer vision community. Therefore, a large number of examples in the literature focused on up-right frontal face detection. Rowley, Bajula and Kanade (1996) and Rowley, Bajula and Kanade (1998) used an artificial neural network trained using a database with 1050 faces. In Rowley's system, the face is formed by a 20 × 20 pixel window which is applied to each pixel in the whole image. If the image has any face about 20 × 20 pixels, it will find it in its first search, and otherwise the window is resized by a factor of 20% (by subsampling) and the search is repeated. This procedure continues until the whole image falls into the 20 × 20 pixel window. It is an exhaustive search process, which is presented in Figure 4.

Yow and Cipolla (1997) show a feature-based method where some local face features are found in the image using geometric and grey level constraints. These features are combined into partial face groups (PFGs) that represents models for eyebrow, eye, nose, and mouth; this idea is presented in Figure 5. All features are then combined in a simple Bayesian network, reducing false positives. Differently

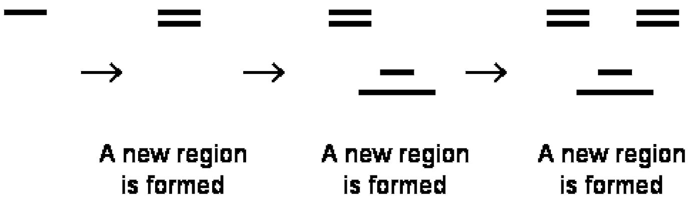

Figure 5 *Face characteristics progressive grouping used in Yow and Cipolla (1997).*

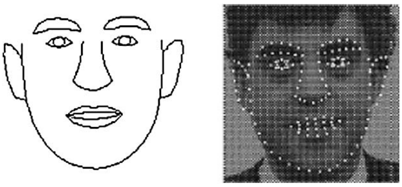

Figure 6 *Cootes and Taylor* (1996) *system: at left the PDM model and at right the face matching process.*

than Rowley the search is not done at every image pixel. Before the search starts, a matched band pass filter is applied to the whole image and the minimal local points are selected as points of interest, where the features can be found. This speeds the process up significantly. At the end, a probabilistic framework assures that low false-positive rate detection will be achieved.

The approach presented in Pham, Worring and Smeulders (2002) uses a forest of three Bayesian networks as a classifier. As in Rowley, Bajula and Kanade (1998) it uses a 20 × 20 window and learns appearance from data, using a data set with frontal images with small variations in rotation and pose. Another approach using Bayesian network is presented in Sebe et al. (2005); the system uses the Bayesian network as a classifier. The network learns a structure from labeled and unlabeled data, once it is learned it receives a vector of values extracted from images and classifies it either as a face or as a non-face. The system's performance is also in the range of 90 to 95%.

Cootes and Taylor (1996) presented a method to model the human face with both shape and gray-level pixel intensity information. The training process used 40 face images, manually marked in their feature points. After that, the algorithm builds a point distribution model (PDM) based on the features characterizing the face. To perform detection and location in new images, they used an active shape model (ASM) that try to fit the face's PDM to the image. Cootes idea is presented in Figure 6; the left image shows the PDM model and the right image shows the matching process. Viola and Jones (2004) presented a different concept for face detection, called integral image, which can be used for frontal face detection. In the integral image, each pixel in the image is represented by the summation of the intensity values of all pixels above and to the left. Once the new representation is

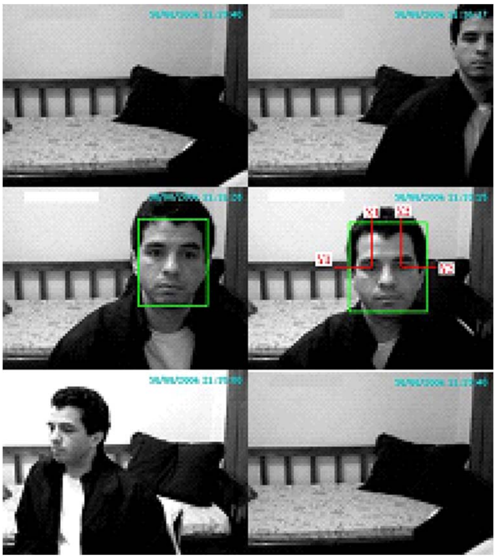

Figure 7 *The operating sequence of the face detection system. The top left shows the background, the top right the user enters in front of the camera. The middle left shows the face detected and the middle right shows the eyes' position. The bottom left shows the user moving away and the bottom right the background again.*

computed the method searches for features in this new image. The method uses features learned from an image database, based on intensity values. The computation of these features is faster using the integral image, increasing the overall performance.

The system presented here for face detection uses a simple image subtraction to start the processing, and then the system uses a geometrical model of the face to compute the features. If the features are found the system computes the eyes' position and send these positions to the eye tracker. While the face is in front of the camera the system keeps tracking the face computing the eyes' position, as presented in Figure 7. The top images show the background (at left) and a user arriving (at right), the middle images show the face detected (at left) and the eyes position (at right), and finally, the bottom images show the user leaving the area (at left) and the background (at right). The face detection system presented here is similar to the system presented in Yow and Cipolla (1997) and it also combines information using a Bayesian network, but our system presents the following unique characteristics:

- The Bayesian network used incorporates the visual operator's reliability.
- Each feature is computed using a different technique allowing a better performance for feature detection.

The face detection system is detailed in the next sections.

3 The model

In an object detection system it is required to describe the object that is the subject of the search or to learn the object from data. When the object is described, a model is created and the system works looking for a match between the image data and the model created. There are several techniques that can be used to describe a model such as production rules, geometrical models, or just a set of parameters. The algorithm will, somehow, evaluate those rules and decide if there is an instance of the object in the data or not. In this section the model used in this work is described and explained.

Notice that the constraints imposed by the problem reduce the possibilities of face variation in terms of size, pose, and orientation. The pose, the camera relative position to the face, must be frontal. If there is a person looking at the screen, its face must be in the up-side view. The orientation can have some variation, but not more than 15 degrees from the vertical axis. About the size, experiments have shown that a person seating in front of a computer, captured with a regular webcam, have a face with about 80 pixels in length and about 120 pixels in height with a 20% variation (considering a 320×240 image). Another limitation is related to the face position in the image. Since the camera is fixed, the face will be shown at the center of the image. Depending on the user's height in front of the screen and its posture, the vertical position of the face can be any. On the other hand, a valid assumption is that the horizontal position of the face will be limited to the central range of the image. Giving some flexibility to the system, it is assumed that a valid face has to be between 20% and 80% of the width of the image.

Processing time is an important issue in this project, because the face recognition phase is supposed to work in real time. In order to avoid the search for features over the whole image the face detection system uses the selective perception (Ballard (1991)) approach, based on a geometrical model that indicates where each feature is expected to appear. The features used in this work are left eye, right eye, skin, and mouth. The geometrical model is presented in Figure 8, which shows the regions in the face where each feature is expected to appear.

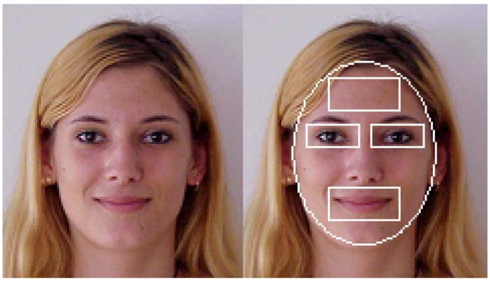

Figure 8 *The face model and the expected regions to measure each feature (left eye, right eye, mouth, and skin).*

The pre-processing step will look for blobs with the size established, inside the central range between 20% and 80% of the image. When a blob with these characteristics is found, the system uses the geometrical model to confirm or not confirm the presence of each feature. The selective perception approach reduces the search space, so the system runs faster and it is capable to analyse the image stream in real time.

Once the features are measured the outcome from the skin colour operator, the mouth operator, and both eye operators are combined in a Bayesian network (for details about Bayesian networks, see Pearl (2001, 1988); Jensen (1996)).

As the network structure is not learned from data, so it is expected to be robust against changes in the dataset. The skin colour operator uses intensity values from the RGB data and, interestingly, these values do not vary too much over different skin colour or races (Vezhnevets, Sazonov and Andreeva (2003)). The mouth operator is a geometrical model using a flexible contour and the eye operator learns to classify eyes from data, but eyes, in general, are quite similar (eyebrowns, pupils, iris, and sclera).

3.1 Fusion of feature information

Once each feature is measured and before the system decides about the presence of a face, it has to deal with two issues:

- How much does it believe in the presence of a feature?
- How does the belief in the presence of a face changes when different features are found, for instance, if the system finds a right eye and a mouth, or skin and mouth?

The technique selected to deal with these two issues was the Bayesian network approach used in Marengoni et al. (2003). The network used is shown in Figure 9. Although the relationship between the features is an important aspect to be considered, because the network is simple and the feature operators are computationally

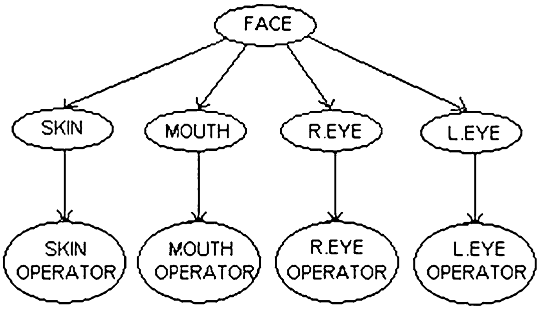

Figure 9 *The Bayesian network model used to combine the feature outcomes after applying the feature visual operators.*

cheap it was decided to remove these relationships links from the network structure.

The network has two layers of nodes for the features: one layer for the nodes representing the features themselves, these nodes (skin, right eye, left eye, and mouth) represent simply the presence or absence of those features in the image; and a second layer for the nodes representing the feature's operators outcome, the operator nodes represent the result (true or false) of the algorithm applied to the image. The links between these two layers represent the reliability of the operator on measuring the presence of the feature in the image ($P(Feature/Feature\ Operator\ Outcome)$) and the links between the feature layer and the root node gives the conditional probability of deciding about face once the feature was found in the image ($P(Face/Feature)$).

The Bayesian network used has face in the root node, which is the node the system checks to find out if a face was detected or not. As there is no way to know the prior probabilities for the presence of a face or not, the prior probabilities for this node were defined as $P(Face) = P(\neg Face) = 0.5$. The network's second layer shows the nodes representing the features that will be measured. In order to determine the conditional probability table for each link, a simple experiment was designed: people were asked to analyse the image set; for each image a person should give a set of answers, if the image had a face or not, and if the face had each of the features or not. Table 1 shows the probabilities found in this experiment.

Using these results the conditional probability table for each feature ($P(Feature/Face)$) was defined. To find the reliability for the operators a second short experiment was performed where each operator was used in a set of images with and without a face, and the conditional probability was computed from the operator's outcome. These conditional probabilities are presented in Tables 2, 3, 4, and 5 for skin, mouth, right eye, and left eye, respectively.

Notice that the skin and mouth operators are quite reliable; they detect the feature presence when it is present and the feature absence when it is not there. The eye operator, on the other hand, is not reliable for detecting the feature presence but it is good on detecting its absence.

The inference in the Bayesian network works as follows: once the image background changes, the system makes a hypothesis for a face in the area changed (blob). The system applies the face model on the blob and, for each area in the

Table 1 *The conditional probabilities $P(Face/Feature)$ for each link in the second layer of the Bayesian network*

Face	*Skin*	*Mouth*	*Right eye*	*Left eye*
YES	35/43	38/43	37/43	32/43
NO	4/31	3/31	3/31	1/31

Table 2 *Conditional probability table for the skin operator* $P(skin/skin\ operator\ outcome)$

	Skin presence	
Operator outcome	YES	NO
YES	33/39	6/35
NO	6/39	29/35

Table 3 *Conditional probability table for the mouth operator* $P(mouth/mouth\ operator\ outcome)$

	Mouth presence	
Operator outcome	YES	NO
YES	28/33	11/41
NO	5/33	30/41

Table 4 *Conditional probability table for the right eye operator* $P(right\ eye/right\ eye\ operator\ outcome)$

	Right eye presence	
Operator outcome	YES	NO
YES	21/41	3/33
NO	20/41	30/33

Table 5 *Conditional probability table for the right eye operator* $P(right\ eye/right\ eye\ operator\ outcome)$

	Left eye presence	
Operator outcome	YES	NO
YES	21/40	3/34
NO	19/40	31/34

model, it applies the corresponding feature operator. Then it takes the feature operator outcomes and feeds them into the Bayesian network leaves nodes propagating the information up, to the root node. At this point the system checks for the probabilities in the root node. If the probability for a face is greater than a threshold, it accepts the hypothesis and sends the information about the eyes' position for the next layer. Otherwise, it waits for other change in the background to repeat the process.

4 Feature operators

The face detection system is composed of: the preprocessing procedure that checks for changes in the background; the Bayesian network to combine the information from the feature operators; and four feature operators, one for the skin detector, one for the mouth detector, and one for each eye. Each of these operators is described in details in this section.

4.1 The skin colour operator

The use of a webcam allow us to easily work with colour images, therefore, one obvious choice is to work with a skin colour operator. This approach has been used in Wu, Chen and Yachida (1994). One problem trying to identify the region of skin colour in an image is to deal with the illumination variations. Looking at the characteristics of skin colour on different races, it is noticed that the skin color presents major variations in intensity, keeping the chrominance invariant. Therefore, the skin colour operator basically transforms the 3-dimensional RGB information into a 2-dimensional RGB normalized space. This plane represents the chrominance variation for the skin colour samples took from images; it is located in a small region of this plane, as shown in Figure 10.

To calculate the standard deviation and average of the skin colour it was used about 200 pixels from 20 images. The decision was made calculating the Euclidean distance between the pixel and the average, and comparing with the standard deviation. If the distance is below the standard deviation, it means the colour is inside the skin region on the chrominance space. Inside the geometrical face model used, the forehead was defined as the region where the skin operator is applied to. Then, the operator computes the total of skin pixels and a threshold is used to decide if the skin is present or absent.

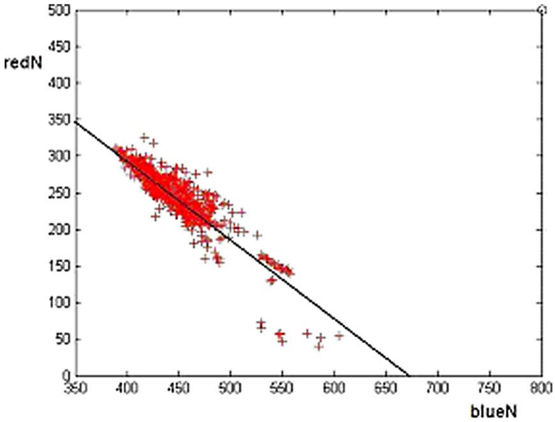

Figure 10 *Skin colour distribution for the test images in the RGB normalized plane.*

4.2 The eye operator

Basically, the eye operator is a feed forward artificial neural network (NN). It operates in the following way: first a 15 × 8 window is cropped from the image and the greyscale pixel values forms a unique 120 elements vector, which is used as the input layer of the NN. The NN was trained to give a response equal to 1 when a vector, formed with an eye image, is present in the input layer. Equally, the response is equal to -1 for non-eye inputs. The NN structure has 120 neurons in the input layer, 2 hidden layers (the first one with 50 neurons and the second one with 25 neurons), and, finally, one neuron in the output layer. The problem working with this type of NN application is to model the false positives. Given an image, any spot, except those that are eyes, work as a false positive. A training procedure was used as described in Rowley, Bajula and Kanade (1996), that picks up in the training images, the examples of false-positives detection. It works as follows:

(1) First the NN is trained with 40 images of eyes using the resilient backpropagation technique and starting with its weights randomly chosen, until it reaches a 100% correct detection.
(2) Now the NN with the weights adjusted in step 1 runs over 20 face images, selecting all spots where false-positive detection occurred.
(3) All those selected spots are cropped from the image and they are used in a new training section starting with the same weights that it had at the end of step 1. The eyes examples are also used in the new training section.
(4) Steps 2 and 3 are repeated until the NN reaches an acceptable level of false-positive detection.

Even after this interactive process, the NN presents an individually false-positive detection rate high enough to compromise the final results. Hence, it was used three NNs, with the same structure but trained with different initial weights, and the operator decides for positive just when all three NNs give a positive response at the same time.

4.3 The mouth operator

The mouth operator is a model-matching or template-matching algorithm, partially based on Cootes and Taylor (1996), where a face flexible model is deformed step by step until it matches the contours of the head and the internal features of the face. In this work, it was built a mouth model made of five points which is used to perform the search for the mouth in the image. After the pre-processing stage, the region of the mouth is known, and the algorithm will run over that region. The model (shown in Figure 11) fits inside any mouth contours, and its points start moving in a defined direction, searching for the contour points. When all points find a contour, the algorithm can analyse the transformation of the model and decide if a mouth has been found.

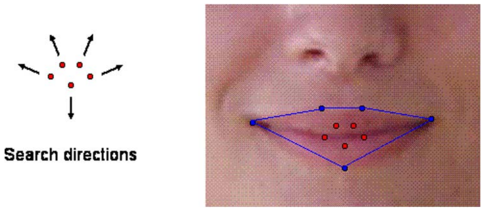

Figure 11 *At left the directions the model moves to search the mouth boundaries. At right the model shown inside a mouth at the start position (the dots) and after finding the mouth contour.*

Just like in the skin colour operator, the colour of the pixel plays an important role in this operator. The mouth contour is defined as being a point where the colour of the pixel changes from mouth class to skin class in the normalized RGB space.

5 Experimental results

Two sets of experiments were performed using the face detection system. For the first set of experiments a face database was created, this database was composed of 119 images divided into two groups: one group of face images (69 images) and another group of non-face images (50 images). This dataset was divided also into two sets: a training set and a testing set. The training set was composed of 74 images (43 face images and 31 non-face images); the testing set had the remaining images. Figure 12 shows a sample of images from the database. The training set was used for training the neural networks for the eye detectors, for the colour computation, for the skin and mouth operators, and for finding the reliability of the feature operators, used for the conditional probability tables in the Bayesian network. For the eye operators, 20 NNs were created, changing initial weights. The networks with the best performance were selected for the right eye operator and the left eye operator. Once the operators were tuned they were used in the database testing set. The results of this experiment are shown in Table 6.

The efficiency column in Table 6 shows the system ability in correctly detecting a face when there is a face in the image and also the ability to detect that there is no face when there is a change in the background but no face is present. Notice that the system's overall performance was above 93.3%. It correctly found 23 faces out of 26 and correctly identify that there was no face in 19 images without a face. Although the number of false positives was small, 7%, it is important to mention that false positives is not a problem for the optical mouse application because if the face detection system triggers the eye tracking system and no eye is found, the optical mouse aborts and gives control back to the face detection system, which will start the search over.

The second set of experiments used a completely new database for the system, the FRI CVL Face Database from University of Ljubljana, Slovenia. No image

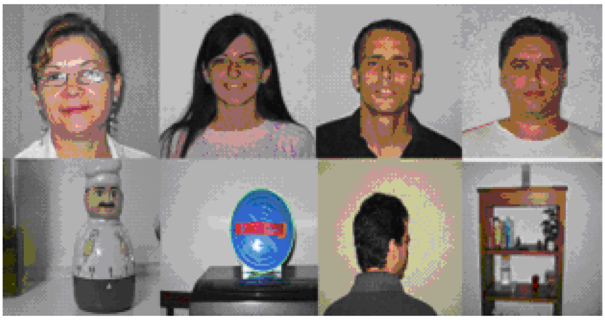

Figure 12 *Sample of images from the dataset: the top row has images from the face group; the bottom row has images from the non-face group.*

Table 6 *Results for the face detection system applied on the database testing set in the face and non-face images groups)*

	Face detection		
Group	YES	NO	*Eficiency*
FACE	23	3	88.4%
NON-FACE	0	19	100.0%

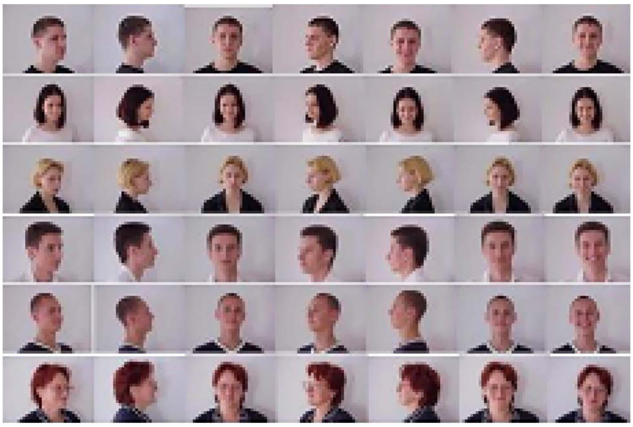

Figure 13 *The FRI CVL Face Database from University of Ljubljana in Slovenia. It has 114 persons each with 7 pictures, three frontal view, two side views, and two partial frontal views.*

from this database was used in the training phase. A sample of the images from the database is shown in Figure 13.

The images with a frontal view were used in this experiment; a total of 342 images were presented to the face detection system. The system correctly found 299

Table 7 *Correct detection rate for each operator and for the face detection system*

Operator	Skin	Mouth	Eye	System
Detection rate	78.2%	75.3%	58.9%	87.4%

faces out 342, a correct detection rate of 87.4%. One important point to add is that no operator by itself was capable of giving this performance, but the combination of operators in the Bayesian network structure ended up increasing the detection rate. Table 7 show the correct detection rate (% of faces correctly detected in the database) obtained by each operator alone and the system's overall performance combining the information from the feature operators.

6 Conclusions and future work

The face detector system presented here is part of an ongoing optical mouse project. The performance obtained by the face detector system shows that it is reliable for the application being developed. The system's performance was 93.3% over 45 images from the data set created for this experiment, containing face and non-face images. When using the face detection system over a new data set containing 342 frontal images, the system was capable to correctly classify 299 images, that is, a detection rate of 87.4% without any additional training.

The survey presented in Yang, Kriegman and Ahuja (2002) shows the performance of several general face detectors, as presented in Table 8. Notice that the face detector system presented here, combining cheap computational operator in a Bayesian network, shows a performance compatible with the best general face detector systems, with a detection rate of 93.3% using our database and 87.4% using a new database.

Although this face detector has been developed to deal with frontal-view faces only, a more general system can be designed using the same concept. With a few changes to the feature operators, and perhaps adding other features, the network could be trained to detect face profiles as well.

The next step in the eye mouse project is to incorporate in the system the capability of tracking the face once it was detected. Starting at the location where the face was detected for the first time, the tracking system will follow the movements of the face and keep passing the eye position to the next layer, the eye tracker.

Another task that would help to evaluate the use of the Bayesian network includes the use of a learning system to learn the network structure such as the BN Constructor Cheng, Bell and Liu (1998). Entering with data gathered from the operators applied to the training set images, the software can search for the network structure which best represents the events. One piece of information this work would demonstrate is how evidence in a node changes the likelihood of a node

Table 8 *Performance of general face detectors from Yang, Kriegman and Ahuja (2002)*

Method	*Reference*	*Detection rate*
Distribution Based	Sung and Poggio (1998)	81.9%
Neural Networks	Rowley, Bajula and Kanade (1996)	90.3%
Naive Bayes Classifier	Schneiderman and Kanade (1998)	91.2%
Support Vector Machine	Osuna, Freud and Girosi (1997)	74.2%
Linear Subspaces	Yang, Ahuja and Kreigman (2000)	89.4%
SNoW Primitive Features	Yang, Ahuja and Kreigman (2000)	93.6%

belonging to another branch. For example, the evidence that an image has a right eye will increase the probability for the left eye. It is not certain whether this kind of causality (linking the eyes) would give a better performance or not.

Another point to explore is the necessity of computing all the features, perhaps, once mouth and left eye are found these should be enough to conclude that there is a face in the image and the other features would not be required.

References

Ballard, D. H. (1991). Animate vision. *Artificial Intelligence* **48** 57–86.

Chai, D. and Ngan, K. N. (1998). Locating facial region of a head-and-shoulders color image. In *Third IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR)* 124–129. Nara, Japan.

Chen, Q., Wu, H. and Yachida, M. (1995). Face detection by fuzzy pattern matching. In *Proceedings of The International Conference on Computer Vision* 591–596.

Cheng, J., Bell, D. and Liu, W. (1998). Learning Bayesian networks from data: An efficient approach based on information theory. Technical Report Department of Computer Science Univ. Alberta.

Cootes, T. F. and Taylor, C. J. (1996). Locating faces using statistical feature detectors. In *Second IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR)* 204. Killington, USA.

Huang, K. S. and Trivedi, M. M. (2008). Integrated detection, tracking, and recognition of faces with omnivideo array in intelligent environments. *Journal on Image and Video Processing* **8** 1–19.

Jensen, F. V. (1996). *An Introduction to Bayesian Networks*. Springer.

Kadoury, S. and Levine, M. D. (2007). Face detection in gray scale images using locally linear embeddings. *Computer Vision and Image Understanding* **105** 1–20.

Mamalet, F., Roux, S. and Garcia, C. (2007). Real-time video convolutional face finder on embedded platforms. *EURASIP Journal on Embedded Systems* **2007**, Article ID 21724.

Marengoni, M., Hanson, A., Zilberstein, S. and Riseman, E. (2003). Decision making and uncertainty management in a 3D reconstruction system. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25** 852–858.

Masip, D., Bressan, M. and Vitri J. (2005). Feature extraction methods for real-time face detection and classification. *EURASIP Journal on Applied Signal Processing* **2005** 2061–2071.

Osuna, E., Freud, R. and Girosi, F. (1997). Training support vector machines: An application to face detection. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition* 130–136.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference*. Morgan Kaufmann. MR0965765

Pearl, J. (2001). Bayesian network. *Handbook of Brain theory and Neural Networks*. MIT Press.

Pham, T. V., Worring, M. and Smeulders, A. W. M. (2002). Face detection by aggregated Bayesian network classifiers. *Pattern Recognition Letters* **23** 451–461.

Rowley, H., Bajula, S. and Kanade, T. (1996). Neural network-based face detection. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition* 203–208.

Rowley, H., Bajula, S. and Kanade, T. (1998). Rotation invariant neural network-based face detection. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition* 38–44.

Scheneiderman, H. and Kanade, T. (2000). A statistical method for 3D object detection applied to faces and cars. In *Proceedings of The International Conference on Computer Vision*.

Schneiderman, H. and Kanade, T. (1998). Probabilistic modeling of local appearance and spatial relationship for object recognition. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition* 45–51.

Sebe, N., Cohen, I., Cozman, F. G., Gevers, T. and Huang, T. S. (2005). Learning probabilistic classifiers for humancomputer interaction applications. *Multimedia Systems* **10** 484–498.

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

Vezhnevets, V., Sazonov, V. and Andreeva A. (2003). A survey on pixel-based skin color detection techniques. In *Proceedings of the 13th Graphicon* 85–92.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision* **57** 137–154.

Wu, H., Chen, Q. and Yachida, M. (1994). Face detection from color images using a fuzzy pattern matching method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21** 53–63.

Wu, H., Yokoyama, T., Pramadihanto, D. and Yachida, M. (1996). Face and facial feature extraction from color image. In *Second IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR)* 345–350. Killington, USA.

Yang, M. H., Ahuja, N. and Kreigman, D. (2000). A SNoW-based face detector. In *Advances in Neural Information Processing Systems* (S. A. Solla, T. K. Leen and K. R. Muller, eds.) **12** 855–861. MIT Press.

Yang, M. H., Ahuja, N. and Kreigman, D. (2000). Mixtures of linear subspaces for face detection. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition (ICAFGR)* 70–76.

Yang, M. H., Kriegman, D. J. and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** 34–58.

Yow, K. C. and Cipolla, R. (1997). Feature-based human face detection. *Image and Vision Computing* **15** 713–735.

Pós Graduação em Engenharia Elétrica
Universidade Presbiteriana Mackenzie
Rua da Consolação 930
São Paulo/SP, 50740–540
Brazil
E-mail: jcandido@mackenzie.br
URL: http://www.mackenzie.br

Pós Graduação em Engenharia Elétrica
e Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie
Rua da Consolação 930
São Paulo/SP, 50740–540
Brazil
E-mail: mmarengoni@mackenzie.br
URL: http://www.mackenzie.br