An Image Processing and Belief Network approach to face detection

Paulo Sérgio de Souza Coelho[1], Cláudio Esperança[1], Antonio Alberto Fernandes de Oliveira[1]

[1]Laboratório de Computação Gráfica,
COPPE - Programa de Engenharia de Sistemas e Computação
Caixa Postal 68511, CEP 21945-970, Rio de Janeiro, RJ, Brasil
psergio, esperanc, oliveira@lcg.ufrj.br

Abstract. This work describes an approach for face detection, which is the first stage of any fully automated human face recognition system. We propose several enhancements to a feature-based approach described by Yow and Cipolla[20] in an attempt to obtain more accurate results. Namely, the attentive feature selection and grouping phases are modified in order to deal with incomplete feature detection while, at the same time, reducing the number of candidates and candidate groups considered.

1 Introduction

Recognizing faces in images is a very common task performed in different contexts by, for instance, security systems controlling the admission to a building or specialized searching systems working on image databases. In fact, the subject received a lot of attention during the last decade and, as a result, a large number of works have been published.

Here we consider that face recognition is the task of comparing a face already detected in an image with those of a data base, possibly obtained under different conditions. The task of identifying if and where faces occur in an image is known as face detection. We consider face detection as a task that must precede face recognition. Some authors, however, consider it a part of the face recognition task. The terms "face search" and "face location" are also commonly used to refer to face detection.

Several different approaches to face recognition have been proposed and we can find in the literature many works where high recognition rates are reported. Some of these techniques take face detection for granted, that is, they assume the existence of a face detection procedure capable of determining with reasonable accuracy all the needed contours and features. Others assume that the images have been taken in conditions which are so constrained that face detection becomes quite easy. In reality, however, face detection is frequently complicated by factors such as background complexity, illumination conditions, image scale and the pose of the face. Even when we have a frontal-parallel view of a face, there are some alternatives to be considered. For instance, the open mouth of a smiling face can be considerably different from a closed mouth. Bangs, eye glasses and moustaches sometimes mess up the result of a detection system that does not consider their presence. In a side view of a face, even when all main features are present, symmetry is lost. In a profile view, some features disappear and the remaining ones may become more important.

In the present state-of-the-art, face detection systems usually do not take all those alternatives into account. Nonetheless, a relatively robust system which supports varying scales, illumination conditions and face orientations was described in Yow c. That system uses a belief network to get high detection rates even for profile images. In summary, the system identifies in the image possible locations of elementary features such as an eye, an eyebrow, a nose or a mouth (we shall use the term candidate to refer to such locations). The belief network is used to estimate not only the probability of a given candidate actually representing an elementary feature, but also to identify what groups of candidates could be considered to represent a face or a part thereof.

In this article we propose enhancements to the so called pre-attentive feature selection and the grouping phases of Yow's approach. We modify the former in order to reduce the number of candidates, and the latter in order to avoid testing a huge number of pairs and quadruplets. We opt to make training more elaborate rather than operating on the image for each pair of candidates which satisfy a simple positional condition. In spite of the geometric constraints the number of such pairs can be very high.

Moreover, incomplete detection may lead to false results. Suppose that instead of having a candidate representing the mouth, we have two candidates which are parts of the mouth. Now, let G be a group where only one of these candidates play the role of a mouth, and let $G^{'}$ be that same group except that both candidates form a mouth. One should expect that the probability of G belonging to a face will be lower than that of $G^{'}$. On the other hand, we cannot join, say, the two eyebrows, although they might have exactly the same characteristics. To address the problems due to incomplete detection, the system attempts to recognize fragment of a face element and to fit them together into a face element candidate.

Finally, making use of simple positional relationships and organizing the data, we can make grouping less computationally intensive. Assuming that the system has already assembled fragments, it will only pair two feature candidates if their relative size match those of the features they represent. The distance between the two feature candidates is treated similarly. Moreover, if two features overlap too much, then they are either merged into a single feature or one of them is discarded. Thus, the number of pairs of feature candidates that have to be tested grows linearly with the overall number of candidates.

The article is organized as follows: in Section 2 all best-known approaches for face detection are overviewed. Our approach is described in detail in Section 3. Section 4 is dedicated to the system implementation and the results obtained so far. Finally, Section 5 contains some final comments and perspectives for further work.

2 Related Work

During the last ten years a large number of strategies for face detection have been proposed. They can be roughly classified in one of the following categories:[1]

2.1 Shape-based Systems

These systems use intensity or color contrast to identify contours which are then compared with the contour of standard facial features.

The best known methods of this category use active contour models (Waite and Welsh [16], Craw et al. [4] and Cootes and Taylor [3]). An active contour is a curve which can be deformed or attracted to an image contour in response to a system of forces. Snakes (see Terzopoulos and Waters [14]) are the most popular of these models. The main drawback of using active contour models is the fact that they require both good initial solutions and an adequate choice of parameters to perform well. This creates difficulties for the detection of facial features. If an active contour with the shape of a mouth is placed close to an eye, then it is possible that it will converge to the eye leading to a false detection. Model energies relative to each standard facial feature can be used to reduce the frequency of that kind of problem. However, this is not sufficient to eliminate the problem completely, specially considering that facial features can take different forms and in consequence the model energy function cannot have a very localized support in the space of contours.

2.2 Feature-based Systems

These systems exploit the fact that the relative position of any two human face elements is considerably fixed. As a consequence, the geometrical relationship between the feature candidates corresponding to these elements in an image is less sensitive to viewpoint changes than intensity or shape. The first stage of such systems is a feature detection step which is crucial to the ultimate quality of the results obtained. Feature candidates detected in the image are then compared with the facial features, usually by means of a statistical measure defined in a space of characteristics. For instance, a very popular approach uses the correlation between windows in the image and feature templates (Sumi and Ohta [12], Zelinksy and Heinzmann [21]). However, since the image of a feature can vary considerably due to pose or illumination variations, its correlation with the feature template can lower down to the point of making the method unappropriate.

Once two different feature candidates are detected, their relative position can be used to check whether they can be part of a face or not. This grouping process is repeated until a sufficient number of different features of the same face is recognized. The bottom-up nature of this technique makes it possible to reject hypotheses at low levels of the grouping hierarchy thus contributing to the efficiency of the approach. They are also a good alternative to deal with situations where some of the face elements are not present, as in a profile view or when the face is partially occluded.

The present work is based on the approach suggested by Yow and Cippola ([20]) and, like it, can be classified as a Feature-based System.

2.3 Pattern-based Systems

Most popular approaches to pattern-based face detection use neural networks to tell whether a given region of an image contains a face. As one can expect, the quality of the results is related to the amplitude of the data used in the training phase. Sung and Poggio [13] used over 1,000 face images and 10,000 images with no faces to train their multi-layer perceptron network. Another well-known technique (see, for instance, Turk and Pentland [15]) consists in projecting the image onto the subspace generated by the eigenvectors associated to the k largest eigenvalues of a pixel corre-

[1] Due to text length restrictions we will only cite some of the most representative works of each class.

lation matrix obtained in the training phase Note that these eigenvectors are images, and not features.

Pattern-based systems have difficulty when dealing with different imaging situations such as partially occluded faces or changes in the face pose or illumination conditions. Such systems can be used in a less restrictive context, but only if the training data includes several images of the same individual taken under slightly different conditions.

2.4 Color-based Systems

Several researchers (Fleck et al. [6], Kjeldsen and Kender [8]) have demonstrated that skin color varies within a narrow strip of the color space. In view of that, detecting pixels having the color of the human skin can be a straightforward way to find image locations where a face must be searched. Region growing techniques can be used to group skin color pixels, and for each of these groups a face hypothesis can be formulated.

Of course, those techniques are independent of face poses or viewpoints. Chen et al. [2, 17] obtained interesting results using an uniform perceptual color space and a fuzzy logic classifier. Using a combination of color and texture, Dai and Nakano[5] were able to obtain a large number of correct face detections. The problem is that the color of an object perceived by a human depends on the light wavelength which varies, for instance, with the time of the day. Moreover, different cameras produce different color values. Hence, if there are no records about the conditions under which an image has been taken, the range of possible colors for the skin becomes too large making the approach less advantageous.

2.5 Motion-based Systems

Most of these systems assume a static background and make use of very simple techniques such as subtracting consecutive frames to reduce the search space. However, if the viewpoint also varies or if there are several moving objects in the scene, then the number of false candidates generated can make this approach totally unproductive. Nevertheless, a motion detector is a powerful tool to confirm face hypotheses.

3 The Proposed Approach

The face model used in the detection step is composed of six facial features: the two eyes and eyebrows, the nose and the mouth. These features were chosen with two objectives: to make the model reasonably independent of the various factors affecting the appearance of a face in a photo, and to guarantee that the model contains a minimum amount of information. Indeed, except when the face is considerably occluded or when the line of vision is almost parallel to the plane of the face, at least four of the selected features do appear in the image. Any such group of four features is called a Partial Face Group (PFG). This property gives the model a certain stability in relation to the factors influencing the aspect of a face in an image. In a frontal view, the presence of several PFGs can be used to increase the confidence of the detection.

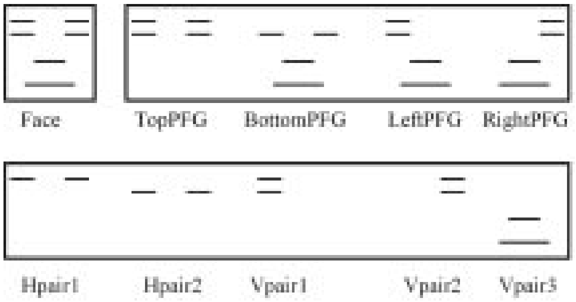

Figure 1: The models used to face and facial features groups.

The recognition procedure is preceded by a training phase where a standard vector of characteristics is associated with each facial element (i.e., nose, mouth, eyes, etc.) and also to fragments of these elements[2]. Once a candidate is identified in the image, the probability that it actually represents a given feature is estimated by computing a statistical measure of similarity between the candidate's vector and that feature's standard vector. We remark that these characteristics do not need to be the same for every feature, although, some of them (mean values, variances) are common to all.

The approach used here is divided into two phases. The first one consists on a search for features whose result is a list of candidate locations. For each pair (candidate-feature) we estimate the probability of the candidate be an instance of the feature. Using these probabilities and geometrical and positional relationships, the system tries to merge candidates associated with fragments of the same face element. After this merging process, the candidates that have a reasonable probability of representing a complete face element are selected. The second phase consists of a process whereby the selected candidates are grouped according to their relative positions. These should

[2] We use the term feature to denote both an element and fragments of an element, while the term characteristic refers to a measure associated to a feature (a mean value, a variance, the aspect ratio, etc.). We call attention to the fact that a fragment of a face element is also refered to as a feature in the text following.

match the relative positions of the correspondig face elements. For instance the candidate corresponding to the nose must be above and not very far to the one corresponding to the mouth. Groups of candidates that do not adjust to each other in a coherent manner are discarded. The result of this phase is a set of candidates corresponding to one or more PFGs, and the probabilities that they are part of a face. These probabilities are obtained by using Mahalanobis distances whose parameters are defined in the training phase for each group.

The context of these two phases are completely distinct. The first one makes use of image processing techniques: differential and Gaussian filters, edge detection methods and histograms for determining thresholds. In the second one, a Belief Network is used to obtain more precise estimates of the probability that a group belongs to a face. That network makes it possible to explore the interrelationship between groups sharing a common candidate. For example, a highly probable pair composed of an eye candidate and an eyebrow candidate may increase the probability of a pair composed of that eye candidate and another one. The characteristics used for a group of face elements can be either dependent on those employed for each face element alone or consider information not taken into account for any element (e.g.: the relative distance between two elements forming a pair).

3.1 Searching Features

The facial elements that comprise our model (eyebrows, eyes, nose and mouth) have a common property. If the image is subjected to a strong low-pass filter, then all these elements – even the nose, if the nostrils appear - become strips with a horizontal dimension much longer than the vertical one. Thus, the search can be oriented in that direction. Following a suggestion given in Yow [20], we have used a filter which is a Gaussian approximation of the partial derivative operator $\left(\frac{\partial^2}{\partial y^2}\right)$ convoluted with a Gaussian filter defined in a window longer in the horizontal direction than in the vertical one (we used a ratio of $3:1$). This filter intensifies the horizontal edges and weakens the vertical ones. The idea is to avoid having to eliminate edges explicitly in function of its direction, which could be complicated.

The resulting filter is

$$F(x,y) = \frac{k}{2\sigma^4\pi}\left(\frac{y^2}{\sigma^2} - 1\right)\ e^{-\dfrac{\left(\frac{x}{3}\right)^2 + y^2}{2\sigma^2}} \qquad (1)$$

for some $k > 0$ [7]. Gaussian Derivative Filters are separable and also scalable and steerable, meaning that they can be expressed as a linear combination of a finite collection (i.e., a base) of similar filters (see Perona [11]). By implementing the filters of such a base, it is possible to obtain information about the orientation and scale of a face. This, in turn, makes it possible to normalize the image with respect to these two aspects.

The standard deviation σ must be chosen according to the size of the features in the image. If nose and mouth are merged into a single feature, them it will not be possible to recognize them. On the other hand, a σ which is too small can generate a huge amount of candidates thus decrease the performance. For 256×256 images of close faces, values of σ between 2 and 3 have produced reasonable results. Once the filter above is applied, the next step is to localize all local maxima in the resulting image. A large number of these maxima is to be expected and, hence, a strict threshold has to be used to select only a fraction of them. Usually, most local maxima correspond to false candidates and several of them are associated whit the same feature. Local maxima which survive the thresholding process - hereafter called points of interest or simply p.o.i.'s - will be pruned later in three further steps. First, we search for edges laying above and below the p.o.i. If it is not possible to find near the p.o.i. an edge whose direction is predominantly horizontal (at an angle with the x-axis less than $\frac{\pi}{6}$, for instance), then the p.o.i. is discarded.

Around each of the remaining p.o.i.'s, we place a bounding box whose horizontal sides are approximations of the closest edges above and below the p.o.i.[3].

From the data contained in that bounding box, we obtain a vector of characteristics that will represent a feature candidate associated with the p.o.i. Using the Mahalanobis distance (see Eq. 6), this vector will be compared with similar vectors obtained in a training phase. As a result of that comparison, the candidate, and in consequence the p.o.i., can also be pruned.

In order to find (almost) horizontal edges around a p.o.i., we filter the image locally using the operator

$$\left(G * \frac{\partial}{\partial y}\right)(x,y) = \frac{\partial G}{\partial y}(x,y)\,. \qquad (2)$$

As we make a local search for each p.o.i., it is possible to use thresholds adjusted to the data in its neighborhood. As mentioned above, if no edges larger than a given minimum length are found, then the p.o.i. is discarded. To avoid that happening too often, chaining techniques are used to try to merge small segments

[3] In fact, we work with a lower bound for the width of the bound box. Boxes which are too narrow contain very little information.

close to each other into a long one. In fact, a complete edge finding procedure is performed locally.

Now, consider a narrow vertical strip symmetric in relation to the column of the p.o.i. Starting at the line of the p.o.i., the system goes upward summing up the horizontal extents of all edges crossing the strip within the current line. This procedure is repeated in the downward direction. When the accumulated total of both searches reaches a given minimum value, the process stops. Now, let p_{lu} (p_{ru}) be the leftmost (rightmost) point of an edge that meets the scanned part of the strip above p.o.i. p.

Let p_{ld} and p_{rd} defined analogously in relation to the part below p. The horizontal dimension (D_H) of the bounding box associated to p can be estimated as

$$\frac{1}{2}\left[(x_{ru} + x_{rd}) - (x_{lu} + x_{ld})\right]. \qquad (3)$$

The vertical dimension (D_V) could be obtained in a similar way. However, problems may arise if, say, p_{lu} is below p. To avoid that, in the calculation of D_V we use the vertical coordinates of the highest points of the edges containing p_{lu} and p_{ru} and the lowest points of the edges containing p_{ld} and p_{rd}.

Consider three images obtained from the original: image (1) is obtained by applying the first degree operator given in 2; image (2) is obtained by applying the second degree operator given in 1, and image (3) is the Hilbert Transform of image (2). We have selected eight measures to form the vector of characteristics of a bounding box. Four or six of them, depending on the feature, are mean values and variances of these images inside the bounding box. The remaining measures are geometrical in nature and also depend on the type of feature being considered.

During the training phase, the image is subjected to the same process up to this point. Then, the user has to indicate the actual bounding box of each of the six face elements. Let Q be the bounding box of a face element F in an image used in the training, and let Q' be one of the bounding boxes obtained when the methodology above is applied to that image. If the area of $Q \cap Q'$ is simultaneously larger than 1/3 of the area of Q and larger than 4/5 of the area of Q', then Q' will be associated with F or a fragment of F. The feature which will be associated with Q' is determined in function of the textures represented in Q'. For instance we consider four possible features related to an eye: the eye itself and three fragments: iris, white and one composed of half the white and the iris. The mouth has even more fragments because the teeth or even a part of the tongue can appear, besides the lips. Now, let Q_1 and Q_2 be two bounding boxes placed one beside the other and associated with fragments, F_1 and F_2.The candidates relative to Q_1 and Q_2 wil be merged if the area of $Q_1 \cup Q_2$ is only a fraction (say 1/5) smaller than the area of its bounding box. If $F_1 = F_2 = F$, then the candidate resulting of the merge will be also associated with F. Otherwise, it will be associated with the feature composed of both F_1 and F_2. If the number of training samples where a given fragment is detected is considered small, then that fragment can be explicitly indicated by the user in some sample images.

For each feature, the system stores the average values of all characteristics associated with it and the correlation matrix of these characteristics. Considering that feature j is represented in m_j training samples then, its correlation matrix is obtained by

$$\sum\nolimits_j = \frac{\sum_{k=1}^{m_j} (x_{j,k} - \overline{x}_j)(x_{j,k} - \overline{x}_j)^T}{m_j} \qquad (4)$$

where

$$\overline{x}_j = \sum_{k=1}^{m_j} x_{j,k} \qquad (5)$$

is the vector composed of the mean values of all characteristics of feature j.

In the detection step, this statistical information is used by the system to perform a last pruning operation on the set of p.o.i.'s. The Mahalanobis Distance M_{ij} between a candidate i whose vector of characteristics is x_i and the set of samples of feature j obtained in the training phase is given by

$$M_{ij} = (x_i - \overline{x}_j)^T \sum\nolimits_j^{-1} (x_i - \overline{x}_j). \qquad (6)$$

The probability of candidate i representing feature j is first estimated as

$$P_{ij} = \begin{cases} \left(1 - \frac{M_{ij}}{\tau_j}\right), M_{ij} < \tau_j \\ 0, \text{ otherwise} \end{cases} \qquad (7)$$

where τ_j is an admission threshold for the jth feature class.

These probabilities are used in the process of merging candidates associated to fragments of the same element in order to validate or reject unions. The candidates resulting of that merging, whose initial probabilities P_{ij} are zero for all face elements are eliminated.

Lastly, four probability values P_{brow}, P_{eye}, P_{nose}, $P_{mouth}$are associated with the remaining candidates, using the equation above. When a group is formed, the candidates play the role of a specific face element. Only the probability relative to that element will be used in the network when processing the group. For instance, if the pair eyebrow-eye is formed, then only the P_{brow} of the upper candidate and the P_{eye} of the lower candidate will be used. These probabilities are also useful to reduce the number of possible combinations which include the group.

3.2 Grouping Features

The second stage of our detection procedure consists of estimating the face as an appropriate subset of the features detected in the first stage. This stage consists of two separate steps: feature grouping and probability estimation. At this point, the original image is not necessary anymore since the list of bounding boxes obtained in the first stage already contains the required information. Each box has an associated characteristics vector and a list of probability values. Each value indicates the degree of certainty that the box corresponds to one of the face elements and is computed using the Mahalanobis distance described earlier.

The grouping of features is done considering the groups shown in Figure 1 and consists of the following steps:

1. Face elements candidates are examined two at a time in order to build pairs. Candidates which do not take part in any pair are discarded.
2. All pair groups obtained in the previous step are also examined pair-wise in order to form PFGs.
3. If any two PFGs have one pair group in common, then they represent a complete face model.

This feature grouping process is similar to that described by Yow [20] which considers geometrical relationships between interest points. However, since we assume that the face orientation is the standard one the evaluation of these geometrical relationships becomes simple. The first condition for a pair of features be considered an acceptable vertical pair is that their horizontal projections overlap significantly. Thus, if $x_M(i)$ and $x_m(i)$ are the maximum and minimum x values of the bounding box Q_i associated with interest point i, then two interest points n and r are considered a vertical pair if

$$\frac{\min\left(x_M(r), x_M(n)\right) - \max\left(x_m(r), x_m(n)\right)}{\left(\dfrac{x_M(r)+x_M(n)}{2} - \dfrac{x_m(r)+x_m(n)}{2}\right)} \qquad (8)$$

is not less than a given threshold value τ_v. Note that this criterium avoids pairing two candidates whose bounding boxes have very different sizes.

When considering horizontal pairs it is not reasonable to test for an overlapping of the vertical projections since the boxes usually have small y dimensions. Instead, the system only tries to evaluate if the boxes are roughly aligned along a horizontal line. Thus, if $y_M(i)$ and $y_m(i)$ are the maximum and minimum y values of the bounding box associated with interest point i, then for two interest points n and r be considered a horizontal pair they must satisfy that

$$\frac{\min\left(y_M(r), y_M(n)\right) - \max\left(y_m(r), y_m(n)\right)}{\min\left(y_M(r) - y_m(r), y_M(n) - y_m(n)\right)} \qquad (9)$$

is not less than a given threshold value τ_h. They must also satisfy a second condition which is:

$$\left(\frac{x_M(r)+x_M(n)}{2} - \frac{x_m(r)+x_m(n)}{2}\right) \geq .5\times$$
$$\max\left\{x_M(r) - x_m(r), x_M(n) - x_m(n)\right\}$$

Let A_i be the area of Q_i and A_{nr}, the area of $Q_n \cap Q_r$. If $A_{nr} \geq 4/5.\ \min(A_n, A_r)$ then either one between n and r is discarded or they are merged into a single element. Moreover the search for a vertical or horizontal pair containing n is limited by a threshold determined in function of the size of Q_n.

Using that methodology, sorting boxes along the x and y axes and testing only candidate pairs which are close in each sorting order it is possible to expect an average cost significantly smaller than that of Yow [18], since it is not necessary to examine all possible pairs.

As part of the grouping process, probabilities for the groups are estimated using the Mahalanobis distance (6). Groups (pairs or PFGs) with very low probabilities are eliminated. . For each new group generated, a vector of characteristics is obtained based on both the characteristics of each feature composing it and the interrelations between these features. Some of the characteristics are mean values and variances determined from the corresponding ones obtained for the individual candidates. Depending on the group we can (a) repeat characteristics of the individual candidates; (b) use weighed combinations of the values of a given characteristic obtained for each individual candidate, or (c) use the result of simple operations involving the characteristics. For instance, consider a pair of eyebrow candidates. It can be expected that the ratio variance/(mean)^2 calculated in the bounding box of one of the candidates is similar to that of the other. So, if the difference of these ratios is large we can reject the pair[4].

[4] However, if it is small we cannot accept it without further analysis.

This search for adequate characteristics is done in the training phase where some natural choices are tried for each group of features. An adequate choice is important, considering that some compression of information must be done when we group features since the dimension of the characteristic vector should not be enlarged.

3.3 Probability Estimation

The probability of the resulting group being a face is estimated with the aid of a Belief Network. Belief Networks are laid out like singly-connected directed acyclic graphs, i.e., there is at most one path between any two nodes. The nodes represent random variables and edges represents conditional dependencies between the linked nodes. In our network (see Figure 2), nodes correspond to facial features or groups of facial features (pairs or PFGs). Associated with each node B there is a data structure containing (a) a boolean variable b that denotes whether that feature was recognized, (b) a probability vector P (one position for each possible boolean value) associated with that feature, (c) two value vectors named λ and π and (d) two message vectors named λ_B and π_B which are used in the message passing mechanism of the network [5]. An edge is implemented by means of a conditional probability table (CPT), where each entry denotes a conditional dependency between the target node and its parents. For instance, the CPT associated with the edge between nodes Hpair 1 and Leftbrow in Figure 2 contains 4 entries corresponding to each possible state of parent nodes Hpair 1 and Vpair 1.

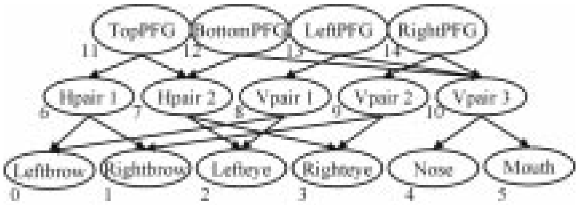

Figure 2: The Singly Connected Network used.

Whenever a node B is instanced (i.e., there is evidence that the associated feature or group of features is present in the image), it modifies the parameters of its parents by sending them λ_B messages and those of its sons by sending them π_B messages. Upon receiving a λ_K or π_J message from one child K or a parent J respectively, a node alters its λ/π value accordingly therefore changing the probability for that node (the equations to perform this changes are given below). If a node receives a λ_K message from one of its sons, the message is propagated to all other sons and all parents. However, if a node receives a π_J message from one of its parents, then this is propagated only to the sons. This is necessary in order to avoid loops in the network. For example, assuming that node 10 was instanced (refer to Figure 2), the following sequence of nodes will be altered: 10, 12, 7, 2, 3, 13, 8, 0, 2, 14, 9, 1, 3, 4, 5.

Assume a node B with two parents P and Q, and a set of sons denoted by $s(B)$. Let b the two possible boolean values of node B (0 to denote false and 1 to denote true). Similarly, let p and q denote the boolean values of nodes P and Q, respectively. Then, the updated probability $P'(b)$ of node B is given by

$$P'(b) = \alpha \lambda(b) \pi(b), \qquad (10)$$

where $\lambda(b)$ and $\pi(b)$ are the λ and π values associated with node B for the boolean value b. α is a normalization factor used to ensure that $P'(0) + P'(1) = 1$.

The λ values of node B depend on the values of the λ messages sent by its sons $s(B)$:

$$\lambda(b) = \prod_{C \in s(B)} \lambda_C(b) \qquad (11)$$

where $\lambda_C(b)$ is the message received by B from its child C. The π values of node B (Eq. 10) are given by

$$\pi(b_i) = \sum_{p=0}^{1} \sum_{q=0}^{1} P(b|p,q)\, \pi_P(p)\, \pi_Q(q), \qquad (12)$$

where $P(b|p,q)$ is the conditional probability of B given its parents P and Q, and $\pi_P(p)$ and $\pi_Q(q)$ are the values of the π messages sent to B by each of its parents P and Q, respectively.

The λ message from a node B to your parent P, $\lambda_B(p)$, is given by

$$\lambda_B(p) = \sum_{q=0}^{1} \pi_Q(q) \left(\sum_{b=0}^{1} P(b|p,q)\, \lambda(b) \right) \qquad (13)$$

and the π message received by B from your parent P, $\pi_P(p)$, is

$$\pi_P(p) = \frac{P'(p)}{\lambda_B(p)} \qquad (14)$$

The Vpair 3 is a different node in the network, because it has 3 parents. If the third node is denoted

[5] Note that λ and λ_B refer to distinct variables, and the same is true for π and π_B. This notation is deriving from [10].

as R, and r indicates any instance of it, then equations 12 and 13 become

$$\sum_{r=0}^{1}\sum_{q=0}^{1}\sum_{p=0}^{1} P\left(b|p,q,r\right)\pi_P\left(p\right)\pi_Q\left(q\right)\pi_R\left(r\right), \qquad (15)$$

$$\sum_{r=0}^{1}\sum_{q=0}^{1}\pi_R\left(r\right)\pi_Q\left(q\right)\left(\sum_{b=0}^{1} P\left(b|p,q,r\right)\lambda\left(b\right)\right) \qquad (16)$$

These equations are sufficient to manipulate the evidence propagation within the network when a new evidence is introduced at one of the nodes. A complete example using these equations to propagate evidence in a singly connected network can be found in Neapolitan [9].

The network used here is a simplification of one which models the face completely. Edges linking the Vpair1 and Vpair2 (nodes 8 and 9) to the topPFG should be present to model the face perfectly[6]. These edges were eliminated because the existence of two paths between the same nodes would make the network much harder to update.

4 Implementation and Results

The algorithm described above was implemented in a RISC 6000 workstation running AIX 3. The software was writen in C and uses an interface developed in TCL-TK. That interface can be seen in the Figure 3. That figure displays a grayscale image with dimensions 305x227. The system assumes that the image contains only frontal-parallel views of faces in standard poses. All problems related to orientation are supposed to have already been solved by using techniques as those described in [20]. To determine an adequate compensation in cases where the orientation is close to the standard ones, steerable scalable filters can be employed in a Preprocessing phase.

The result of the Image Processing phase are highly dependent on the set of filters used to search the points of interest associated to the features. These filters are a simple Gaussian on x and a Gaussian derivative on y. Several alternatives have been tried. For close views of a face the best results have been obtained using variances (σ) in the range $[2, 2.5]$. The best value within this interval depends on the size of the face. This range was used not only for searching features but also to search edges. The value of σ also influences the feature vectors of all points of interest. In relation to the mask used to implement those filters, the best results concerning to the detection of features, have been obtained when the ratio between its horizontal and vertical size is 3.

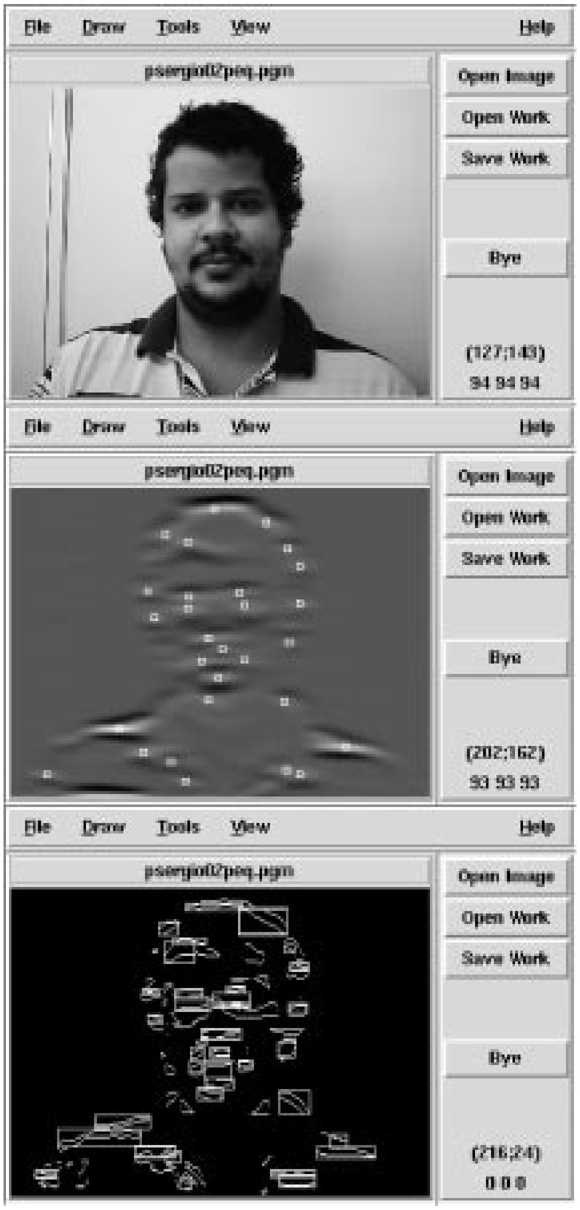

Figure 3: The Interface displaying three stages for an image.

Near the border of the mask the Gaussian Filters are somewhat distorted to make the sum of all mask coefficients be one or zero. Each local maximum of the result of filtered image with (1) represent a feature candidate. Figure 3-middle displays the result obtained by applying this filter to the image in Figure 3-top. The variance used is $\sigma = 2.5$, and the local maxima displayed are those among the 20% higher. So, the 171 local maxima initially obtained are then reduced to 34.

[6] Note in Figure 1 that Vpair1 and Vpair2 should be considered sons of topPFG.

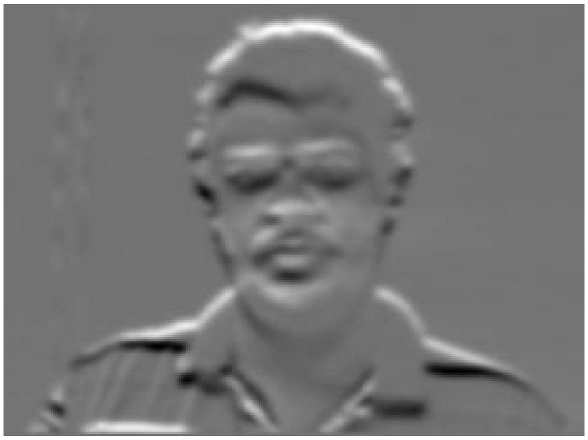

Figure 4: The filter output to edge detection.

For each one of the them, a local search for horizontal edges is performed. If that search succeeds we must have two almost horizontal edges closed to the point. One above it and another below. These edges serve to determine a bounding box around the point. If the search for edges is unsuccessful, the point is discarded.

For searching edges the original image is filtered with (2). In the resulting images the edges are associated to both maximum and minimum values (see Figure 4). Mean values indicate points where the variation in y is low. Figure 3-bottom contains the result of all local searches for edges performed, in the case in study. Some interest points have been suppressed for not having horizontal edges nearby.

The points that have resisted so far are them grouped in pairs. Those pairs will be later subjected to a filtering process and the surviving ones will be grouped in PFGs. For each face feature, (eyebrows, eyes, nose and mouth), all Vpairs and Hpairs and the Partial Face Groups (TopPFG, BottomPFG, Left and RightPFG) characteristic vectors are determined. Once a PFG is identified the network is activated. Some results obtained by it are pictured in Figures 5 and 6. In Figure 5 two successful case are shown, while Figure 6 reports two failures. Both of them can be attributed to large bounding boxes in the region around an eye/eyebrow.

5 Conclusion

In this work we develop a system to perform face detection using a Bayesian Network. The system performance and the accuracy of the network are still being tested. The main concern is to improve the preprocessing phase (Image Processing, Feature Detection and Grouping). Possible extensions are: a) the introduction of new features representing glasses, moustaches, open and closed mouths, b) integrating the information obtained from a sequence of frames and c) treating RGB images.

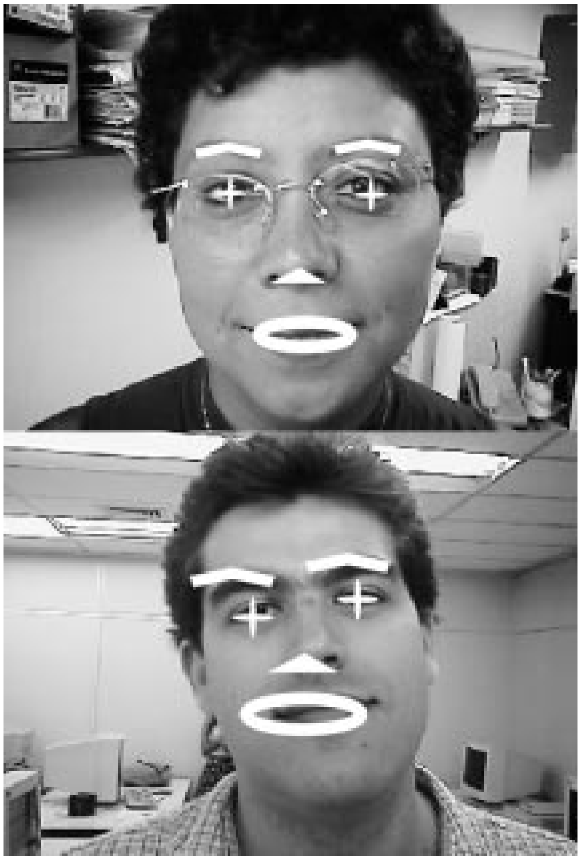

Figure 5: Some detection suceeded.

References

[1] J. Canny. A computational approach to edge detection. IEEE Trans. Patt. Anal. Machine Intell., 8 (6):679–698, 1986.

[2] Q. Chen, H. Wu, and M. Yachida. Face detection by fuzzy pattern matching. In Proc. 5th Int. Conf. On Comp. Vision, pages 591–596. MIT, Cambridge, MA., 1995.

[3] T. F. Cootes and C. J. Taylor. Locating faces using statistical feature detectors. In Proc. 2nd Int. Conf. On Auto. Face and Gesture Recog., pages 204–209. IEEE Comp. Soc. Press, 1996.

[4] I. Craw, D. Tock, and A. Bennet. Finding face features. Proc. 2nd European Conf. on Computer Vision, Italy, pages 92–96, 1992.

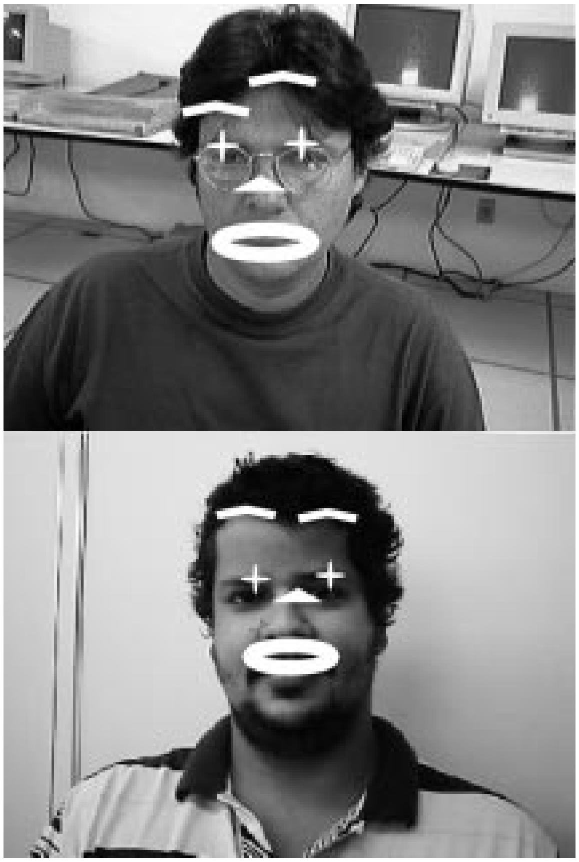

Figure 6: Some false detection ocurred.

[5] Y. Dai and Y. Nakano. Face-texture model-based on sgld and its application in face detection in a color scene. Pattern Recognition, 29 (6):1007–1017, 1996.

[6] M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. In Proc. 4th European Conf. On Comp. Vision, volume II, pages 593–602. Springer-Verlag, 1996.

[7] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. IEEE Trans. Patt. Anal. Machine Intell., 13 (9):891–906, 1991.

[8] R. Kjeldsen and J. Kender. Finding skin in color images. In Proc. 2nd Int. Conf. On Auto. Face and Gesture Recog., pages 312–318. IEEE Comp. Soc. Press, 1996.

[9] R. E. Neapolitan. Probabilistic Reasoning in Expert Systems: Theory and Algorithms. John Wiley and Sons, New York, 1988.

[10] J. Pearl. Probabilist Reasoning in Intelligent Systems. Morgan Faufman, 1988.

[11] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. In G. Sandini, editor, Proc. 2nd European Conf. On Comp. Vision, pages 3–18. Springer-Verlag, 1992.

[12] Y. Sumi and U. Ohta. Detection of face orientation and facial components using distributed appreance modeling. In Proc. Int. Workshop on Auto. Face and Gesture Recog., pages 254–259, Zurich, 1995.

[13] K. K. Sung and T. Poggio. Learning human face detection in cluttered scenes. In J. Hartmonis and J. Van Leeuwen, editors, Computer Analysis Fo Images and Patterns, pages 432–439. Springer-Verlag, New York, 1995.

[14] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. IEEE Trans. on Patt. Analy. and Machine Intell., 15 (6):569–579, 1993.

[15] M. Turk and A. Pentland. Eingenfaces for recognition. J. Cognitive Neuroscience, 3 (1):53–63, 1991.

[16] J. B. Waite and W. J. Welsh. An application of active contour models to head boundary location. In Proc. British Machine Vision Conf., pages 407–412. Oxford, 1990.

[17] H. Wu, Q. Chen, and M. Yachida. An application of fuzzy theory: Face detection. In Proc. Int. Workshop on Auto. Face and Gesture Recog., pages 314–319, 995.

[18] K. C. Yow. Automatic Human Face Detection and Localization. PhD thesis, Department of Engineering, University of Cambridge, 1998.

[19] K. C. Yow and R. Cipolla. Towards an automatic human face localization system. In Proc. 6th British Machine Vision Conf., Vol. 2, volume 2, pages 701–710, 1995.

[20] K. C. Yow and R. Cipolla. Feature-based human face detection. Image and Vision Computing, 15 (9):713–735, 1997.

[21] A. Zelinksy and J. Heinzmann. Real-time visual recognition of facial gestures for human-computer interaction. In Proc. 2nd Int. Conf. On Auto. Face and Gesture Recog., pages 351–356. IEEE Comp. Soc. Press, 1996.