ELSEVIER

Image and Vision Computing 15 (1997) 713–735

image and vision COMPUTING

Feature-based human face detection

Kin Choong Yow*, Roberto Cipolla

Department of Engineering, Speech, Vision and Robotics Group, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

Received 2 August 1996; accepted 20 January 1997

Abstract

Human face detection has always been an important problem for face, expression and gesture recognition. Though numerous attempts have been made to detect and localize faces, these approaches have made assumptions that restrict their extension to more general cases. We identify that the key factor in a generic and robust system is that of using a large amount of image evidence, related and reinforced by model knowledge through a probabilistic framework. In this paper, we propose a feature-based algorithm for detecting faces that is sufficiently generic and is also easily extensible to cope with more demanding variations of the imaging conditions. The algorithm detects feature points from the image using spatial filters and groups them into face candidates using geometric and gray level constraints. A probabilistic framework is then used to reinforce probabilities and to evaluate the likelihood of the candidate as a face. We provide results to support the validity of the approach and demonstrate its capability to detect faces under different scale, orientation and viewpoint. © 1997 Elsevier Science B.V.

Keywords: Human face detection; Perceptual grouping; Feature integration; Evidence propagation; Belief networks

1. Introduction

With the advancement in computer and automated systems, one is seldom surprised to find such systems applicable to many visual tasks in our daily activities. Automated systems on production lines inspect goods for our consumption, and law-enforcement agencies use computer systems to search databases of fingerprint records. Visual surveillance of scenes, visual feedback for control, etc., all have potential applications for automated visual systems.

One area that has grown significantly in importance over the past decade is that of computer face processing in visual scenes. Researchers attempt to teach the computer to recognize and analyze human faces from images so as to produce an easy and convenient platform for interaction between human and computers. Law-enforcement can be improved by automatically recognizing criminals from a group of suspects. Security can also be reinforced by identifying that the authorized person is physically present. Moreover, human facial expressions can be analyzed to direct robot motion to perform certain secondary, or even primary, tasks in our routine work requirements.

In the daunting task of human face processing, face detection is one of the most important problem to be solved. It is a pre-requisite for automatic face recognition and expression analysis. Most automatic face recognition algorithms have either assumed that the face has been cropped from the image (Craw et al. [6]; Turk and Pentland [31]), or they have assumed some constraints about the face and/or background such that the face detection process becomes trivial (Chow and Li [4]).

This task is certainly not trivial when the background is complex, the illumination is varied, and the pose of the face not fixed. Though many successful approaches have been attempted towards face detection and localization, the assumptions and the constraints made in these approaches are still too restrictive, making an extension of the algorithm to more general cases a difficult task. As such, face detection still remains a problem that demands greater attention and further research efforts.

2. Related work

There are a few distinct approaches to face detection. The top-down model-based approach assumes a different face model at different coarse-to-fine scales. For efficiency, the image is searched at the coarsest scale first. Once a match is found, the image is searched at the next finer scale until the finest scale is reached. Some of the work using this

* Corresponding author.

PII S0262-8856(97)00003-6

approach were reported by Yang and Huang [33], and Lanitis et al. [13]. In general, only one model is assumed in each scale (usually in the fronto-parallel view) and thus it is difficult to extend this approach to multiple views.

The bottom-up feature-based approach searches the image for a set of facial features and groups them into face candidates based on their geometrical relationship. Leung et al. [14], Sumi and Ohta [27], and Yow and Cipolla [35] reported work using this approach. Though this approach can be easily extended to multiple views, it is unable to work well under different imaging conditions because the image structure of the facial features vary too much to be robustly detected by the feature detectors.

A texture-based approach was reported by Dai et al. [8]. Faces are detected by examining the spatial distribution of the gray-level information in the subimage (using Space Gray Level Dependency (SGLD) matrices proposed by Haralick [10]). This is again not easily extensible to multiple viewpoints.

The neural network approach detects faces by subsampling different regions of the image to a standard-sized subimage and then passing it through a neural network filter. Recent work was reported by Sung and Poggio [28], and Rowley et al. [22]. The algorithm performed very well for fronto-parallel faces but performance deteriorates when extended to different views of the face. It is still not possible to extend the algorithm to detect faces in profile views.

The colour-based approach labels each pixel according to its similarity to skin colour, and subsequently labels each subregion as a face if it contains a large blob of skin colour pixels (Chen et al. [3]; Dai and Nakano [7]). It can cope with different viewpoint of faces (Chen et al. [3]) but it is sensitive to skin colour and the face shape.

Motion-based approaches use image subtraction to extract the moving foreground from the static background. The face is then located by examining the silhouette (Trew et al. [29]) or the colour of the differenced image (Schiele and Waibel [26]). This approach will not work well when there are a lot of moving objects in the image.

3. About image evidence

So what can we learn from the attempts of these various researchers? Lanitis et al.'s approach is able to locate faces very well because they make use of gray-level image profile in addition to edge information in their statistical shape model (active shape model – Cootes et al. [5]) of the face. Sung and Poggio's method works very well too because almost every pixel in a 19×19 subimage is used to evaluate the output, and many of these pixels encode spatial and gray-level information. However, these methods at present are limited to fronto-parallel views. Leung et al.'s feature-based approach seems to give the flexibility of extension into different viewpoints, but the lack of evidential support in the feature detection process curbed its success.

On the other hand, we can see why Leung et al.'s, Sumi and Ohta's method did not perform as well. The system is dependent on too few image features, which cannot be extracted robustly due to image noise or noise in the feature detector. Leung et al. use the response from a set of steerable-scalable filters to find facial features, and Sumi and Ohta use template matching to identify eyes. In both these cases the evidence for a feature to be present comes largely from the response of the filter or the correlation output. As a result, there is a lack of evidence to support the hypothesis of a face and therefore the performance of the algorithm is affected.

Human vision is very robust because we made use of a large amount of evidence from the visual image that is formed in our retina. Some of these evidence include edges, corners, lines, bars, blobs, intensity, shape, texture, colour and even motion. These types of evidence are well-used by vision researchers in their various approaches. Another form of evidence that is less well-exploited is that of contextual evidence, i.e. the knowledge that certain features occur in the vicinity of other features. For example, we know that eyes occur in pairs. So, when we find an eye in the image, the existence of this eye is evidence for the existence of the other eye.

In this paper, we adopt a bottom-up feature-based approach which has the flexibility to be extended to different scale, orientation and viewpoint of faces in the image. A large amount of geometric, spatial and gray-level measurements are used for robustness. Due to the use of many different types of image evidence, we can reduce the strictness of the requirement in each piece of evidence (e.g. threshold level of edge detection filter response, etc.). This makes the algorithm more robust to noise and occlusion, without creating too many false positive candidate faces.

4. The face model

We always need a model of the object in any object recognition task. Leung et al. [14] model the face as a statistical graph consisting of 5 points, namely the eyes, nostrils and nose tip. The advantages of choosing these points are clear: these points are 'interior' points on the face (as opposed to 'exterior' points lying on the face boundary which are easily influenced by background clutter). These points are also 'rigid', i.e. the inter-feature geometry is not easily deformed by different facial expression or different personal identity. However, the disadvantage of using these points are that they span only a small area on the face (thus a large margin of error).

A model of an object in terms of low level image features (such as edges, corners, etc.) is always very difficult to use because the image structure changes very drastically in different images due to changes in scale, image noise, quantization noise and illumination variations. As such, models of explicit shape (e.g. deformable template

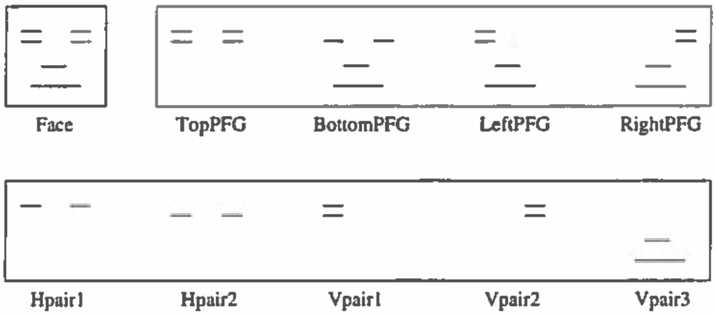


Fig. 1. The face model and the component face groups.

models – Yuille et al. [36]), only work well in high resolution and relatively noise-free images. However, a model of the object described in terms of higher level features (such as a face described in terms of eyes, nose and mouth), is usually quite stable and robust.

We therefore model the face as a plane with 6 oriented facial features (namely, the eyebrows, the eyes, nose and mouth). In addition, the 'cheek' regions (regions under the eyes and to the left or right of the nose and mouth positions), must be relatively feature-free and edge-free. This face model has the advantage of using 'interior' points such as Leung et al.'s [14], and yet it spans a larger area over the actual face, thus making the detection more robust and reliable.

Also, due to occlusion or missing features (eyebrows, usually), we need to decompose the face model into components consisting of 4 features, which are common occurrences of faces under different viewpoints or different identity. These groups are called Partial Face Groups or PFGs (Yow and Cipolla [34]). These PFGs are further subdivided into components consisting of 2 features (horizontal and vertical pairs – Hpair and Vpair) (Fig. 1) for the purpose of perceptual grouping and evidence propagation.

In order for feature detection to be robust we have to use image features that are invariant to changes in scale and illumination intensity. We observe that at low resolutions, all the 6 facial features will appear only as dark elongated blobs against the light background of the face. And since edges are illumination invariant to a large extent, we model the 6 facial features as pairs of oriented edges as shown in Fig. 2. The image is smoothed before the feature detection process so that any high-resolution features will take the form of the lower resolution ones. The vertical edges in the eye and nose model are only used to provide evidence in labelling the facial feature and is not an important criteria in the detection of the feature.

The distinction between facial features and image features should be made clear. In this paper, we define facial features as high-level entities which are present on our faces in accordance to our intuitive idea of the components of a face. Examples are eyes, nose, ear and mouth. We define image features as the low-level entities which we can find from a digital image (e.g. edges, corners, gray-level of pixels and regions). The term features can be used to mean either or both facial features and image features.

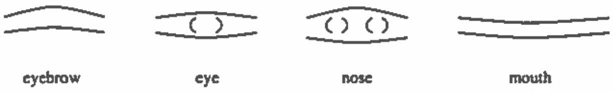


Fig. 2. The facial feature models.

5. Perceptual grouping

It is obvious that with such a low resolution model, there will be lots of false positive feature candidates. We therefore propose a perceptual grouping framework that groups these feature candidates into faces using geometrical, gray-level and spatial information. Feature candidates that cannot be grouped will be discarded.

Perceptual organization is a phenomenon in human vision in which we are able to immediately detect relationships such as collinearity, parallelism, connectivity, and repetitive patterns among image elements. It has been extensively studied by investigators in psychology and computer vision. Excellent surveys of these works can be found in Lowe [15] and Palmer [19].

The Gestalt laws of organization (Koffka [11]; Kohler [12]) states the common rules by which our visual system attempts to group information. Some of these rules includes proximity, similarity, common fate, continuation and closure. Many good perceptual grouping algorithms (Sarkar and Boyer [25]; Mohan and Nevatia [17]) make use of such principles for effective grouping and high performance.

Triesman [30] also proposed a two stage model of perception. The first stage, which is described as pre-attentive perception, extracts image information into points and regions of interest, which directs the attention of processing efforts of the next stage. The second stage of perception, the attentive stage, will perform grouping, comparison, evaluation and reasoning activities based on the detection and identification of meaningful object groups in the image.

We will model our face detection process as a two stage model of perception based on Triesman. The first stage operates on the raw image data and produce a list of interest points from the image, indicating likely location of facial

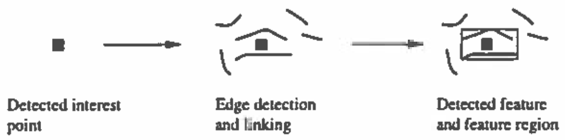

Detected interest point
Edge detection and linking
Detected feature and feature region

Fig. 3. Preattentive feature selection process.

features. The second stage will examine these interest points, group them based on Gestalt principles and label them accordingly to knowledge acquired from training data. The labelled features are further grouped based on model knowledge of where they should occur with respect to each other.

5.1. Preattentive feature selection

The preattentive feature selection is performed in two steps. First, a list of interest points is found from the image using spatial filtering. As we pointed out earlier, at a coarse scale the 6 facial features each resembles a dark elongated bar on a light background. Hence, these features can be found by smoothing the input image and then filtering the image using a matched bandpass filter. A suitable filter will be that of a second derivative Gaussian, elongated at an aspect ratio of 3:1 (Yow and Cipolla [35]). Local maxima in the response will indicate the presence and the location of such structures in the image.

Next, the edges around each interest point are examined. Edges are linked based on their proximity, and similarity in orientation and strength. A standard boundary following algorithm (such as that given in Ballard and Brown [1]) will suffice. Assuming that the face is vertical, we look for almost horizontal edges above and below the feature point. If the orientation is not known, we can look for the existence of two roughly parallel edge segments with opposite polarity on both sides of the interest point. If such a point is found, we flagged it as a facial feature point. We further define the extent of the feature region by drawing a box around the two edges. Fig. 3 illustrates this process.

Measurements of the region's image characteristics (such as edge length, edge strength, gray-level variance) are then made and stored into a feature vector $\mathbf{x}$. From the training data of the facial features, e.g. 'eyebrow', we obtained a mean vector μ_{brow} and covariance matrix Σ_{brow} which define the class of valid 'eyebrow' feature vectors in a n-dimensional space, where n is the number of components defining the feature vector $\mathbf{x}$.

A facial feature candidate i is a valid facial feature j if the Mahalanobis distance $\mathcal{M}_{ij}$ of the feature vector $\mathbf{x}_i$ is within an admission threshold τ_j from the class mean μ_j, i.e.

$$\mathcal{M}_{ij} < \tau_j, \text{ where } \mathcal{M}_{ij} = (\mathbf{x}_i - \mu_j)^{\mathrm{T}} \Sigma_j^{-1} (\mathbf{x}_i - \mu_j). \tag{1}$$

This is repeated for all the 4 classes of facial features, namely, eyebrow, eye, nose, and mouth. If the facial feature does not belong to any of the 4 classes, it is discarded from the list.

There is a significant advantage in using the Mahalanobis distance. The Mahalanobis distance takes into account the variance of the individual parameters in the feature vector. If we use Euclidean distance, and when one of the parameters has a large variance, the Euclidean distance of valid members from the class mean will be large. If the Mahalanobis distance is used instead, this will not be the case because the effect due to that parameter is scaled down by its own variance. This is desirable because the effects of the other parameters will then not be subsumed by the one with large variance, leading to a large reduction in the number of false positives.

5.2. Attentive feature grouping

After obtaining a set of feature points and the associated feature region, these feature regions are then actively grouped using our model knowledge of the face. Single features are grouped into vertical and horizontal pairs, pairs are grouped into partial face groups, and partial face groups are grouped into face candidates (Fig. 4).

For each level k of grouping, a set of n_k measurements is made of the component features and stored into a n_k-dimensional vector. This vector is then projected into its n_k-dimensional class space, which was determined from measurements obtained from faces in training data. The Mahalanobis distance $\mathcal{M}_{ij}$ of this feature vector is then evaluated and used to determine its membership in the class.

This grouping process is effective in removing false positives because a large number of geometric and gray-level measurements are used to determine its validity. In particular, the edge and spatial information about the new region formed that is not part of the components itself

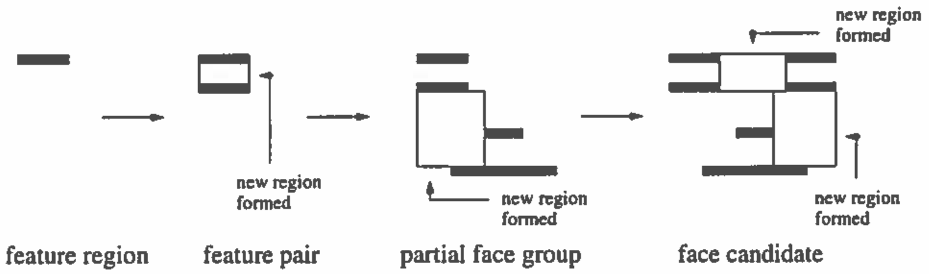
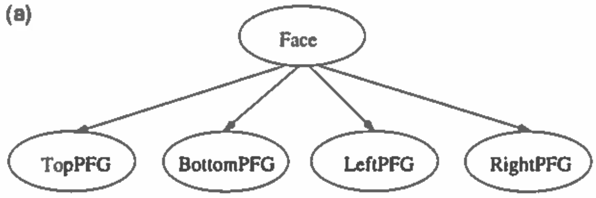


Fig. 4. Attentive feature grouping process.



(b)

TopPFG	T	F
P(TopPFG | Face)	0.6	0.4
P(TopPFG | ¬Face)	0.1	0.9

Fig. 5. (a) A belief network. (b) Conditional probability table (CPT).

(Fig. 4) proved to be the most effective. The effectiveness of this grouping process in not detecting too many false positives is due mainly to the examination of edge and intensity information in these spatial regions.

One important advantage of this bottom-up approach is that though the spatial region to be analyzed gets larger at higher levels, there are fewer of these regions to process. As a result, the processing time is kept small throughout the whole algorithm.

For each additional piece of evidence sought from the image, an additional dimension is used in the feature vector for classification. The more evidence we use from the image, the larger the dimension of the feature vector. In the neural network approach used by Sung and Poggio [28], almost every single pixel in a 19 × 19 subimage is used, leading to a 283-dimensional space. Though this encodes a large amount of image evidence, the class space occupied by valid members can be highly nonconvex, leading to great difficulty in selecting true candidates.

We therefore have to choose measurements based on which features are significant in the image. By examining the features in a large number of images, the measurements that are found to be significant include:

1. the ratio of feature lengths (obtained from edge linking) to the size of the image;
2. the ratio of feature lengths to other feature lengths;
3. the aspect ratio of a feature region;
4. the ratio of inter-feature distances;
5. the difference in orientation between features;
6. the number of directional edge pixels in a region (normalized to the size of the region);
7. the ratio of edge strengths in a region to edge strengths of facial features;
8. the mean gray level of a region (normalized to intensity distribution);
9. the variance in the gray-level distribution of a region.

6. Probabilistic framework

The perceptual grouping framework enables us to reject grossly incorrect groupings of face candidates. Still, we have to deal with a reasonable number of false positive faces which cannot be effectively removed by using detection thresholds in the previous section. We thus propose a probabilistic framework to assign and propagate probabilities among the facial features and facial groups so that we will achieve a high confidence rate for true positive faces.

Bayesian networks, which are also known as belief networks, are directed acyclic graphs, with nodes representing random variables and arcs signifying conditional dependencies specified by conditional probabilities. Bayesian networks do not assume independence among features, they encode the dependencies among features (Russell and Norvig [24]).

An example of a belief network is shown in Fig. 5(a). The nodes with arrows pointing away from them are the parents of those which the arrows are pointing to. This encodes the dependency between the nodes. Each node can take either of 2 values, True or False, and has a conditional probability table (CPT) associated with it (Fig. 5(b)).

The entries in the CPT describe the conditional probability of each value of the variable, given each possible combination of the values of the parent nodes. For example, in Fig. 5, the conditional probability of the node TopPFG being False given that the parent node Face is True is 0.4.

These set of entries can be estimated directly by using the statistics of the set of examples (Russell et al. [23]). The crucial value in the belief network is the prior probability of the root node (the ‘face’ node in this case), and this is often hard to estimate. Certainly, the choice of an appropriate prior depends on the complete space of hypothesis. We may assume an uniform prior for our case.

In our previous approaches [34,35], we used a belief network composed of 4 child nodes, one for each of the 4 partial face groups (Fig. 5(a)). This was shown to be highly effective for fronto-parallel view of faces because all 4 PFGs can be detected in this view, giving a large amount of evidence for true face candidates. However, for profile views, the probability of the face remained low because only one PFG can be found in the image.

To overcome this, we propose a new belief network structure, using the facial features as child nodes instead of the PFGs (Fig. 6). The belief network now has 6 child nodes instead of 4. Profile view of faces will thus have 4 pieces of evidence (facial features) out of 6, instead of 1 (face group)

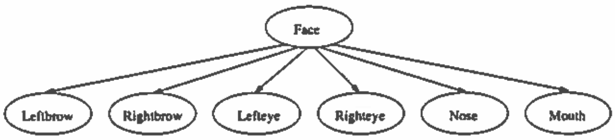


Fig. 6. The belief network used in our approach.

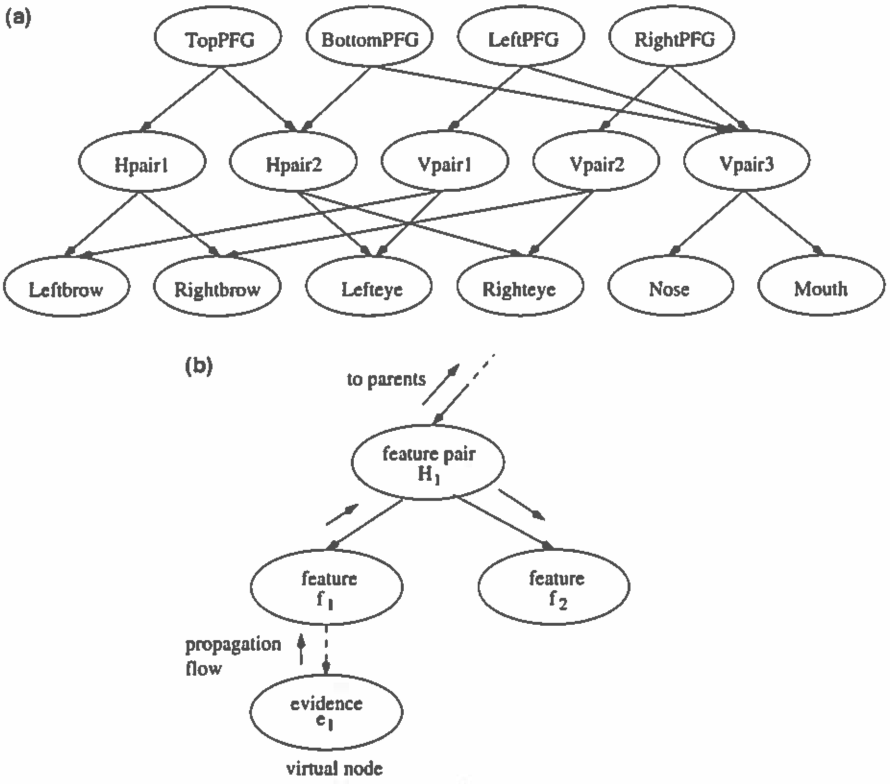


Fig. 7. (a) Reinforcement belief network. (b) Virtual nodes.

out of 4 previously. This leads to a better capability of detecting profile views of faces.

So how do we update and improve the probabilities of these child nodes using model knowledge? As mentioned earlier, one source of evidence that is often overlooked is the presence of a neighbouring feature (e.g. presence of another eye next to an eye candidate). To harness this extra piece of evidence, we build a second belief network (Fig. 7(a)) to reinforce the belief of each feature based on the presence of neighbouring features.

When evidence for a facial feature becomes available, a virtual node is created (the 'evidence' node) and instantiated, allowing the evidence, specified in the form of a probability, to propagate through the entire network and update all the other nodes (Fig. 7(b)). The resulting effect is a large increase in the probabilities of the feature candidates which are true facial features.

We use a propagation algorithm for singly connected networks given by Pearl [20] which does not make any unfounded assumption of the conditional independence of the system. In Pearl's algorithm, each node when instantiated with a piece of evidence will modify its parent or child nodes by sending λ or π messages to them. In addition, each node has a λ and a π value which are modified by these λ and π messages. The main difference between this propagation algorithm and the one for trees (used in our previous work [34,35]) is that nodes in a singly connected network can have more than one parent. Our belief network structure in Fig. 7(a) clearly requires this.

Suppose a node B has two parents A and D, and a set of child nodes $s(B)$ where each child node C is a member of the set $s(B)$, $C \in s(B)$. Also, let B have k possible values b_i, $i = 1,\ldots,k$, A have m possible values a_j, $j = 1,\ldots,m$, and D have n possible values d_p, $p = 1,\ldots,n$. The updated probability $P'(b_i)$ of a node B for the value b_i is then given by

$$P'(b_i) = \alpha\lambda(b_i)\pi(b_i), \qquad (2)$$

where $\lambda(b_i)$ and $\pi(b_i)$ are the λ and π values associated with node B for the value b_i. α is a normalizing constant so that all the probabilities b_i sum to one.

The λ value of a node B, $\lambda(b_i)$ with a set of child nodes $s(B)$ is given by

$$\lambda(b_i) = \prod_{C \in s(B)} \lambda_C(b_i), \qquad (3)$$

where $\lambda_C(b_i)$ is the λ message from the child C to node B for the value b_i. The π value, $\pi(b_i)$, is given by

$$\pi(b_i) = \sum_{j=1}^{m} \sum_{p=1}^{n} P(b_i \mid a_j, d_p)\pi_B(a_j)\pi_B(d_p), \qquad (4)$$

where $P(b_i \mid a_j, d_p)$ is the joint conditional probability of node B given its parents A and D, and $\pi_B(a_j)$, $\pi_B(d_p)$ are the π messages from the parents A and D respectively.

The λ message from a node B to one of its parents A, $\lambda_B(a_j)$, is given by

$$\lambda_B(a_j) = \sum_{p=1}^{n} \pi_B(d_p)\left(\sum_{i=1}^{k} P(b_i \mid a_j, d_p)\lambda(b_i)\right) \qquad (5)$$

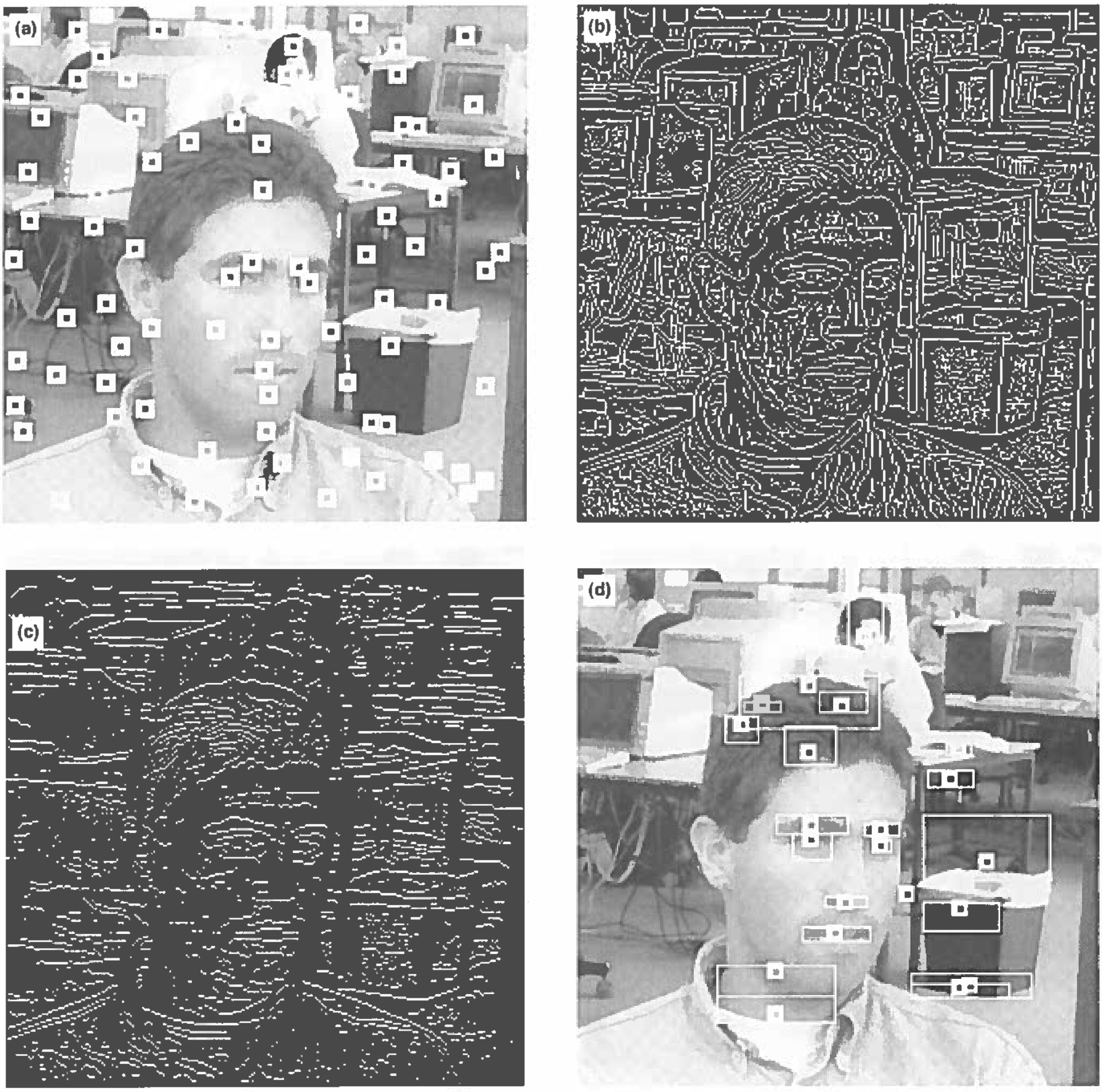


Fig. 8. (a) Interest points obtained from matched bandpass filtering (81 points). (b) Canny edge detection with zero threshold. (c) Linked edges of approximately horizontal orientation. (d) Feature regions detected (21 points).

and the π message received by a node B from its parent A, $\pi_B(a_j)$, is

$$\pi_B(a_j)=\frac{P'(a_j)}{\lambda_B(a_j)}. \tag{6}$$

In the case of a node with more than two parents (e.g. Vpair3), the equations are a straightforward extension of the above. For example, if a node B has another parent E, with r possible values, then eqns (4) and (5) will become

$$\pi(b_i)=\sum_{s=1}^{r}\sum_{j=1}^{m}\sum_{p=1}^{n}P(b_i \mid a_j,\, d_p,\, e_s)\pi_B(a_j)\pi_B(d_p)\pi_B(e_s), \tag{7}$$

$$\lambda_B(a_j)=\sum_{s=1}^{r}\sum_{p=1}^{n}\pi_B(e_s)\pi_B(d_p)\left(\sum_{i=1}^{k}P(b_i \mid a_j,\, d_p,\, e_s)\lambda(b_i)\right). \tag{8}$$

Using these equations, we are able to propagate evidence through the network when evidence for one of the nodes is found. A node when instantiated will send λ messages to its parents, and π messages to its children. In turn, the parent node will send new λ messages to its parents, and π messages to its other children. A child node receiving a π message will only send new π messages to its own

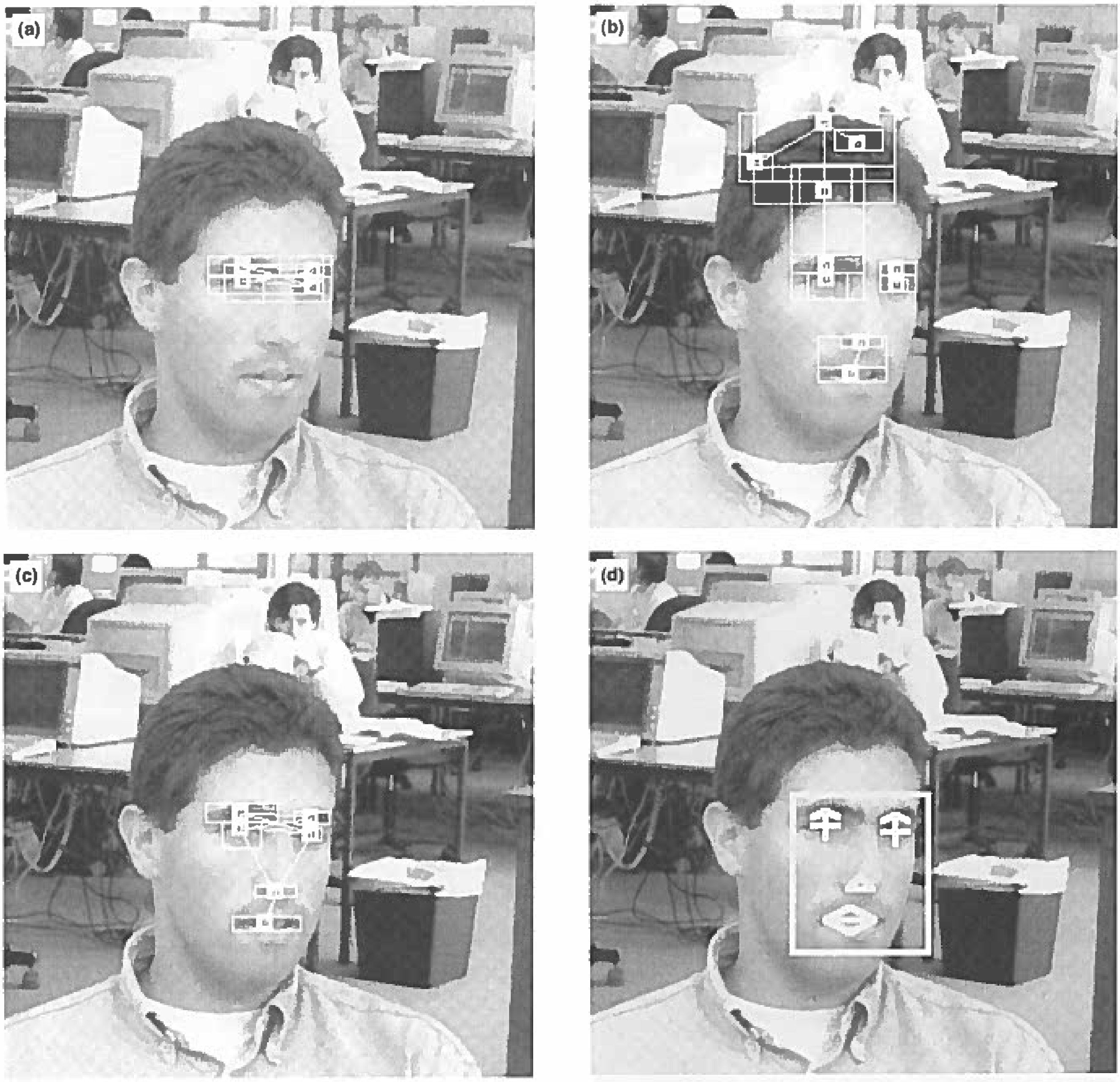


Fig. 9. (a) Horizontal pairs (4 pairs). (b) Vertical pairs (7 pairs). (c) Partial Face Groups (1 top, 2 bottom, 1 left, 1 right). (d) Face candidates (1 face).

children. In this way, nodes which are conditionally dependent on the node that was instantiated will all be updated. An example illustrating the use of the above equations to propagate evidence in a singly connected network is given in Neapolitan [18].

The evidence for each facial feature or face group i is related to its Mahalanobis distance, $\mathcal{M}_{ij}$, and the admission threshold for the jth feature class, τ_j, by:

$$P_i = \begin{cases} \left(1 - \dfrac{\mathcal{M}_{ij}}{\tau_j}\right), & \mathcal{M}_{ij} < \tau_j, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Each facial feature that is detected is assigned 4 probability values, P_{brow}, P_{eye}, P_{nose} and P_{mouth} using the above equation. When a higher level group is formed, only the probability of the corresponding feature is propagated. For example, if a vertical brow-eye pair (Vpair1) is formed from two facial features, only P_{brow} of the upper feature and P_{eye} of the lower feature is propagated. Likewise, only these values are updated in the propagation process. As a result, only true positive faces are updated to a high confidence level.

7. Preliminary implementation and results

We implement the described algorithm making the

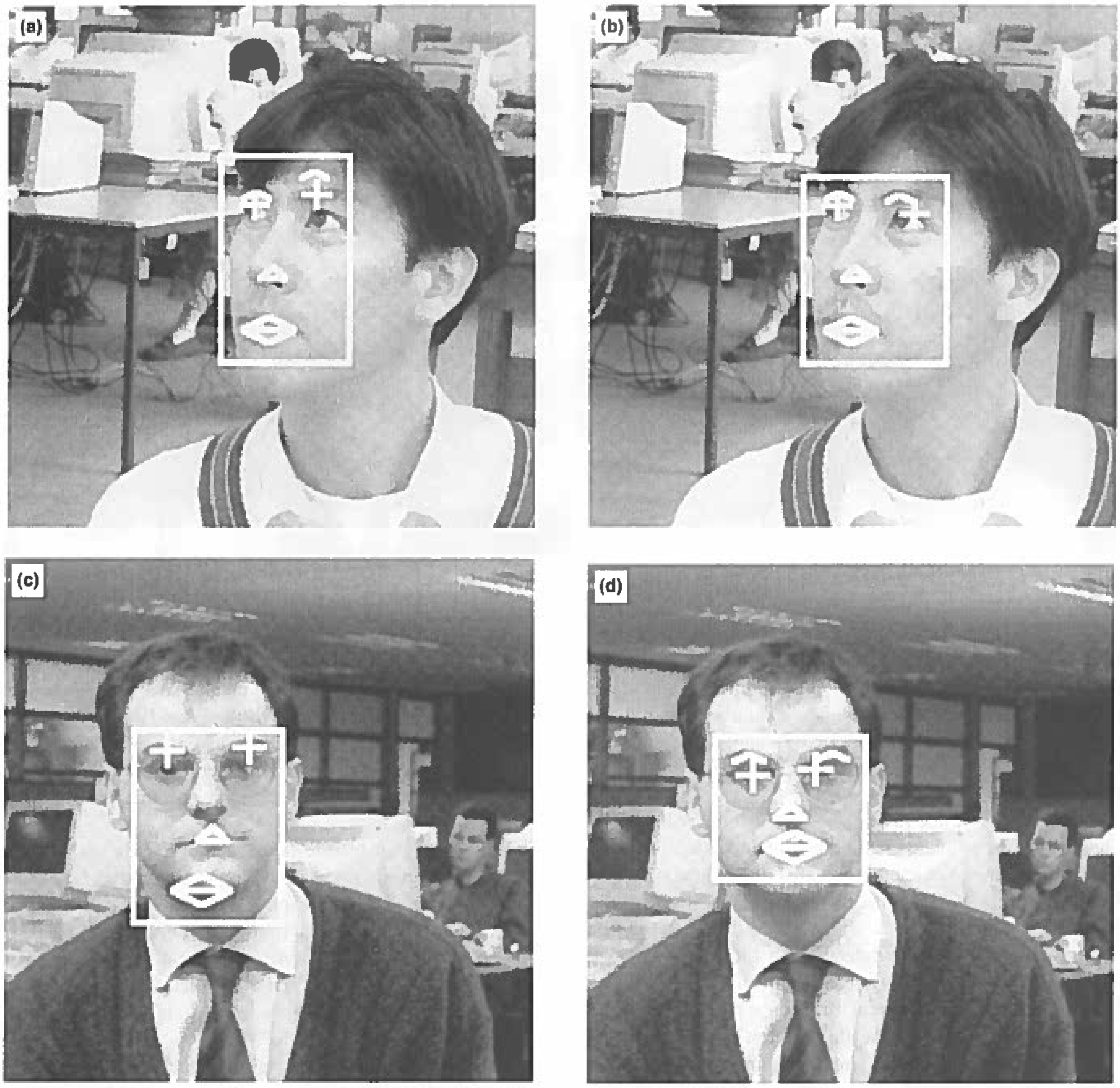


Fig. 10. (a),(b) Face candidates found for subject 1. (c),(d) Face candidates found for subject 2.

assumption that the orientation of the faces are vertical and the viewpoint of the faces are fronto-parallel. This allows us to look at the intermediate results and evaluate the performance and robustness of the algorithm in the simplest case. The scale parameter is specified by the user as the algorithm is run on each image. We will extend the algorithm in the next few sections and show how it can cope with variations in scale, orientation and viewpoint.

7.1. Learning the parameters of the feature class space and conditional probabilities

For this preliminary implementation, we create a database of 100 images by taking 10 images of 10 different subjects under different scale and slightly different viewpoint. Of these 100 images, 40 will be used as a training set and the remaining 60 used as a test set.

We first convolve the image with a matched bandpass filter (σ varying from 1.0 to 3.0 depending on the size of the face in the image) for detecting dark bars against a light background, which is essentially a spatial filter with a second derivative of Gaussian in one direction, and a Gaussian in the orthogonal direction. The aspect ratio of the filter is elongated to 3:1 for better orientation selectivity. Since the scale is specified by the user, and the orientation of the face is assumed to be vertical, we use only a single

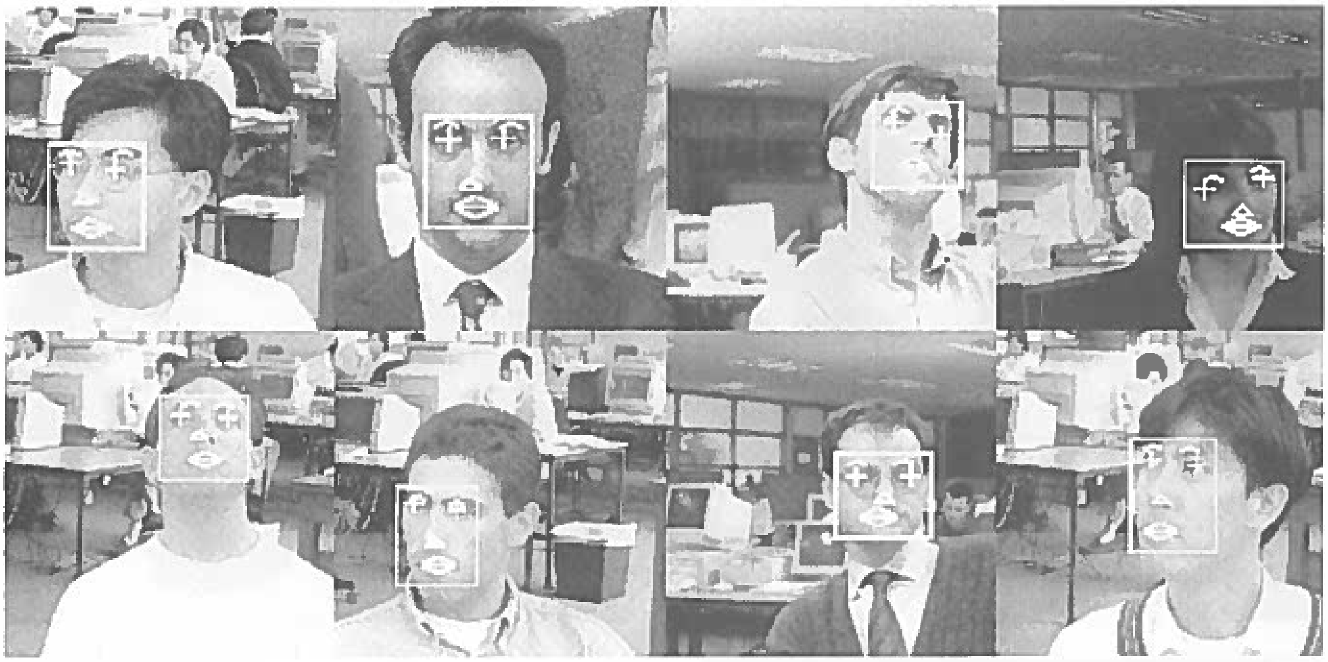


Fig. 11. Result of face detection on various test images.

filter at the specified scale and orientation. If scale and orientation are unknown, we can use a family of such filters at different scale and orientation, and examine the output of each. This family of filters can be efficiently implemented using steerable-scalable basis filters (Perona [21]; Freeman and Adelson [9]).

Local maxima in the response are then found which give a list of interest points which indicate the location of possible facial features. We examine the list of local maxima and determine the 6 points which correspond to the 6 facial features in our face model.

We then perform edge detection on the image using a Canny edge finder with hysteresis threshold set to zero. This will output all edges in the image and will ensure maximum robustness against illumination variations.

A local window around the interest point is then searched for edges in the expected orientation and polarity. The size of the search window is the same as the size of the Gaussian derivative filter used to obtain the interest points. These edges are then linked to form chains. We use a boundary following algorithm given in Ballad and Brown [1] to link the edges. The search continues to edges not in the local window and these edges are linked if they have similar orientation, polarity and strength as the neighbouring edges (Gestalt laws).

For each facial feature (eyebrow, eye, nose and mouth), each feature group (Hpair and Vpair) and partial face group (Top, Bottom, Left and Right), we make measurements of aspect ratio, feature length ratio, etc. (Section 5.2) based on the positions of the 6 feature points that we have marked. The mean and covariance of these measurements are then obtained over the 40 training images and stored as model data.

The perceptual grouping part of the algorithm is then run on the 40 training images with these model parameters. The frequency of occurrence of each feature and each

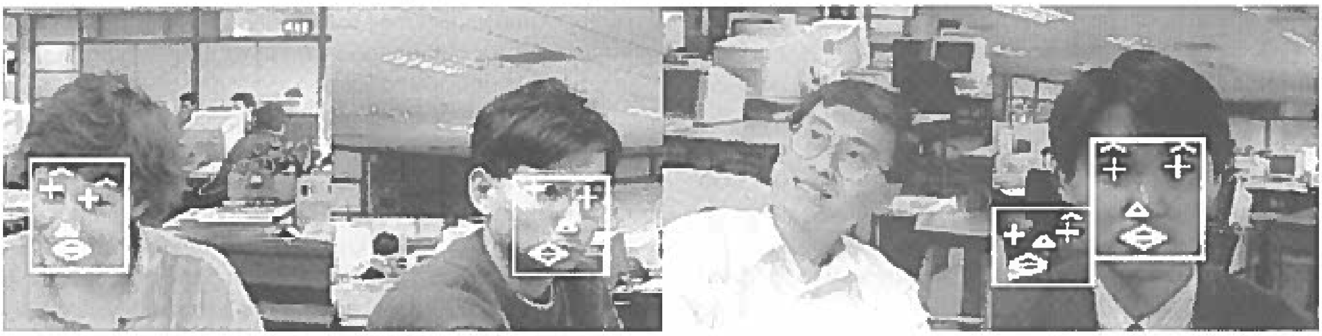


Fig. 12. Some unsuccessful cases.

component face group are measured and entered into the conditional probability table. For example, if each of the 40 training images contain a face each, but we detected 35 topPFGs correctly and 3 topPFGs falsely, we set:

$$P(\text{TopPFG} \mid \text{Face}) = \frac{\text{number of true topPFGs detected}}{\text{total number of faces}} = \frac{35}{40},$$

$$P(\neg\text{TopPFG} \mid \text{Face}) = 1 - P(\text{TopPFG} \mid \text{Face}) = 1 - \frac{35}{40},$$

$$P(\text{TopPFG} \mid \neg\text{Face}) = \frac{\text{number of false topPFGs detected}}{\text{total number of topPFGs detected}} = \frac{3}{(35+3)},$$

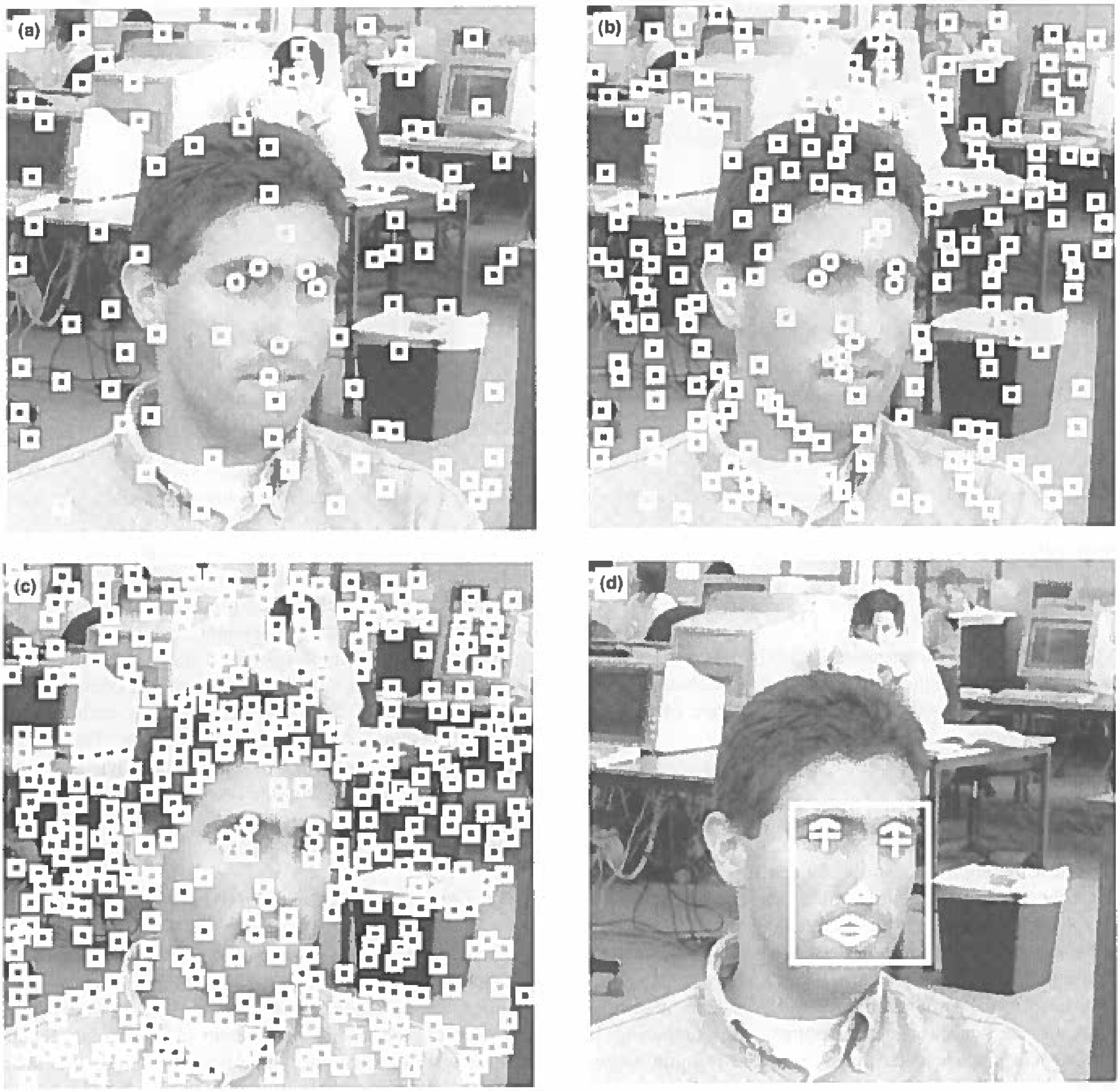


Fig. 13. Varying the scale of the preattentive filter. The facial feature candidates detected by the preattentive feature selection stage are shown. Those feature candidates that correspond to the true facial features are drawn as a circle instead of a square. (a) $\sigma = 3.0$ (81 points). (b) $\sigma = 2.0$ (177 points). (c) $\sigma = 1.0$ (332 points). (d) Face detected (same for all 3 cases of σ).

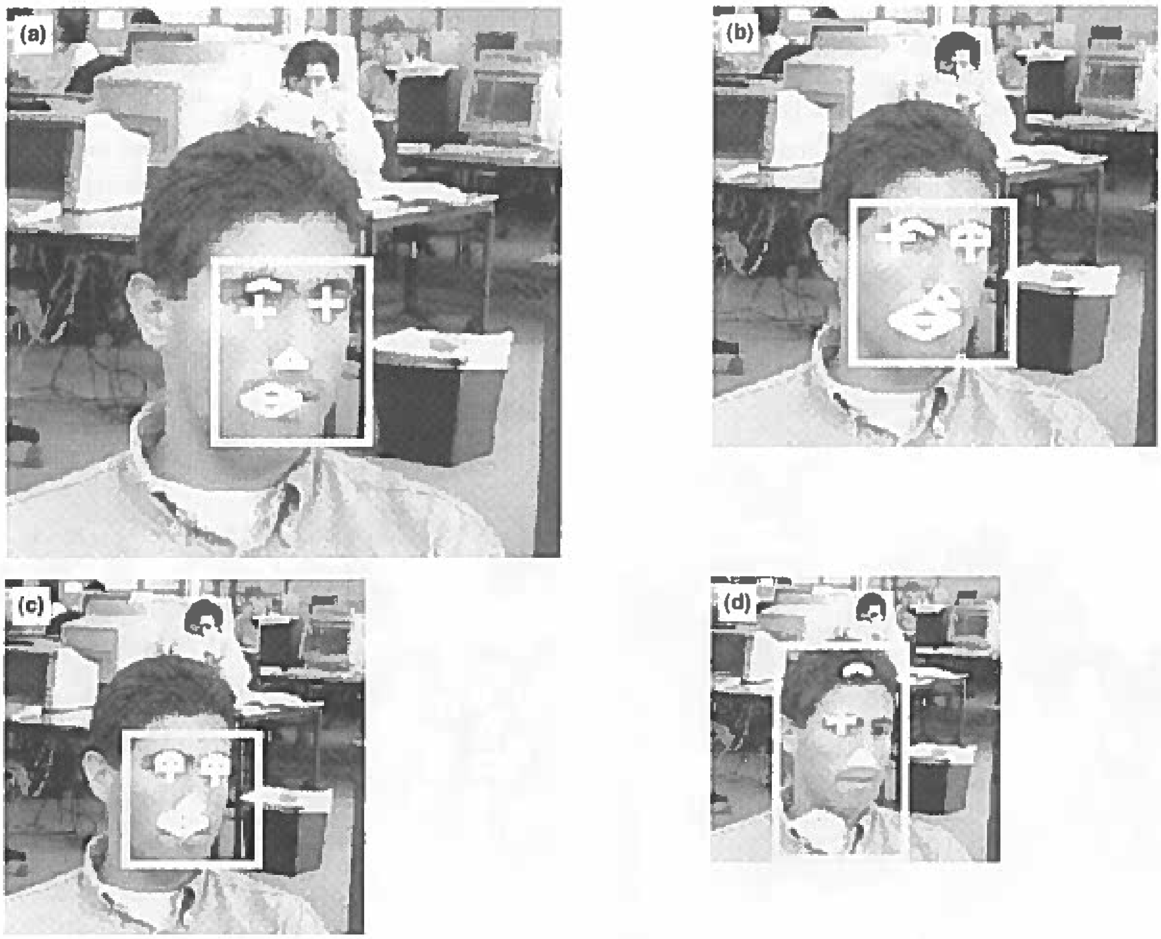


Fig. 14. Varying the size of the face image. (a) Percentage size = 80%. (b) Percentage size = 60%. (c) Percentage size = 40%. (d) Percentage size = 20%.

$$\begin{aligned} P(\neg\text{TopPFG} \mid \neg\text{Face}) &= 1 - P(\text{TopPFG} \mid \neg\text{Face}) \\ &= 1 - \frac{3}{(35+3)}. \end{aligned}$$

$P(\text{TopPFG} \mid \text{Face})$ is not necessarily 1.0 in the set of training images because some facial features may be occluded due to variations in viewpoint, leading to an absence of the top partial face group.

7.2. Feature detection and perceptual grouping

We test the algorithm on the 60 test images, specifying the scale of the preattentive filter for each image. We convolve the image with the preattentive filter, and search for local maxima and edges in the same way as described in the training stage.

By making the same geometric and gray-level measurements of the various interest points and comparing it with the class space determined from the training stage, we obtain a list of facial feature points for the attentive grouping stage.

The results after verification with each feature class is shown in Fig. 8.

The list of feature candidates is then examined to form pairs, and each horizontal pair and vertical pair is further examined to form partial face groups. If any two partial face groups have some component features that are the same, they are combined to form a face candidate (e.g. if a bottom PFG and a left PFG are found, we combine them to give a 6-feature face candidate). If not, each PFG by itself will become a 4-feature face candidate. The validity of each feature group and face candidate that is formed is verified by comparing the geometric and gray-level measurements of each group with the model data. The results for the perceptual grouping stage is given in Fig. 9.

7.3. Evidence propagation and Bayesian classification

Each facial feature that is detected is assigned 4 probability values, P_{brow}, P_{eye}, P_{nose} and P_{mouth}. These probabilities are assigned using eqn. 9. If the Mahalanobis distance of the facial feature in a particular feature class is greater than the admission threshold, the facial feature is given a probability value of zero for that feature class.

After the perceptual grouping process, each face candidate will have between 4 to 6 features associated with it. A

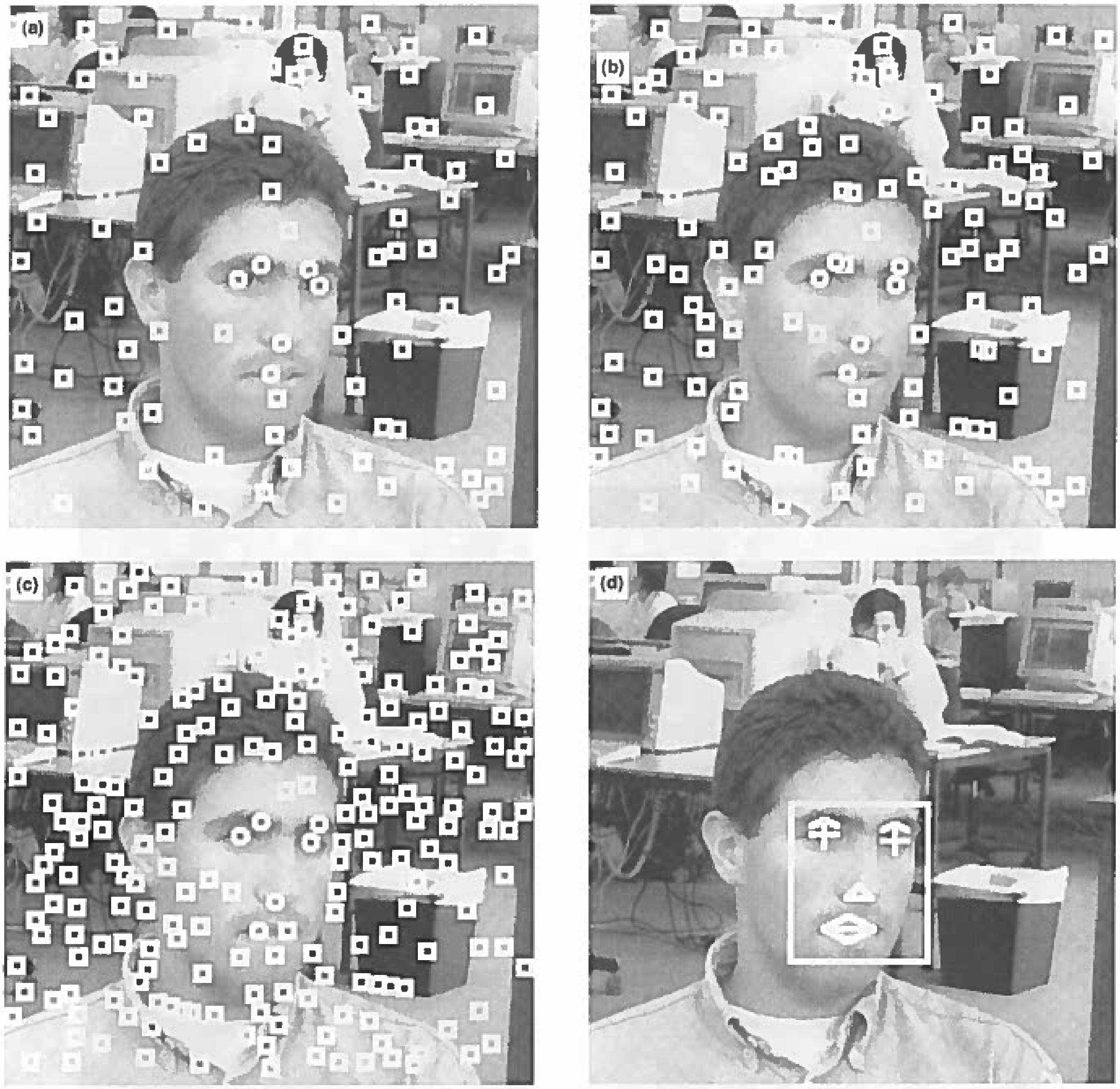


Fig. 15. Varying the aspect ratio of the preattentive filter. (a) Aspect ratio = 3:1 (81 points). (b) Aspect ratio = 2:1 (110 points). (c) Aspect ratio = 1:1 (201 points). (d) Face detected (same for all 3 cases of σ).

reinforcement belief network is initialized for each face candidate and virtual nodes are created for each facial feature that is found in the process.

Fig. 10 shows the face candidates for 2 subjects found by the perceptual grouping process. As these faces cannot exist simultaneously because they overlap, the face with the highest probability will be selected among all the overlapped ones.

For subject 1, in Fig. 10(a), the top PFG is not found in the process and so the computed probability of the face candidate is lower. Moreover, since the hypothesized eye location (on the right) is actually a brow, the image evidence that is propagated in this case is actually P_{eye} which is very low compared to P_{brow} in Fig. 10(b). The probabilities of the two face candidates of subject 1 are 0.6578 and 0.9255, respectively.

For subject 2, only the bottom PFG is found in the first case. The probabilities of the two face candidates of subject 2 are 0.5045 and 0.9486 respectively. Clearly, without the use of the probabilistic framework and the reinforcing of evidence from all the other facial features, the difference between the true and false positive candidates will be very close, thus making it very difficult to successfully reject the false candidates.

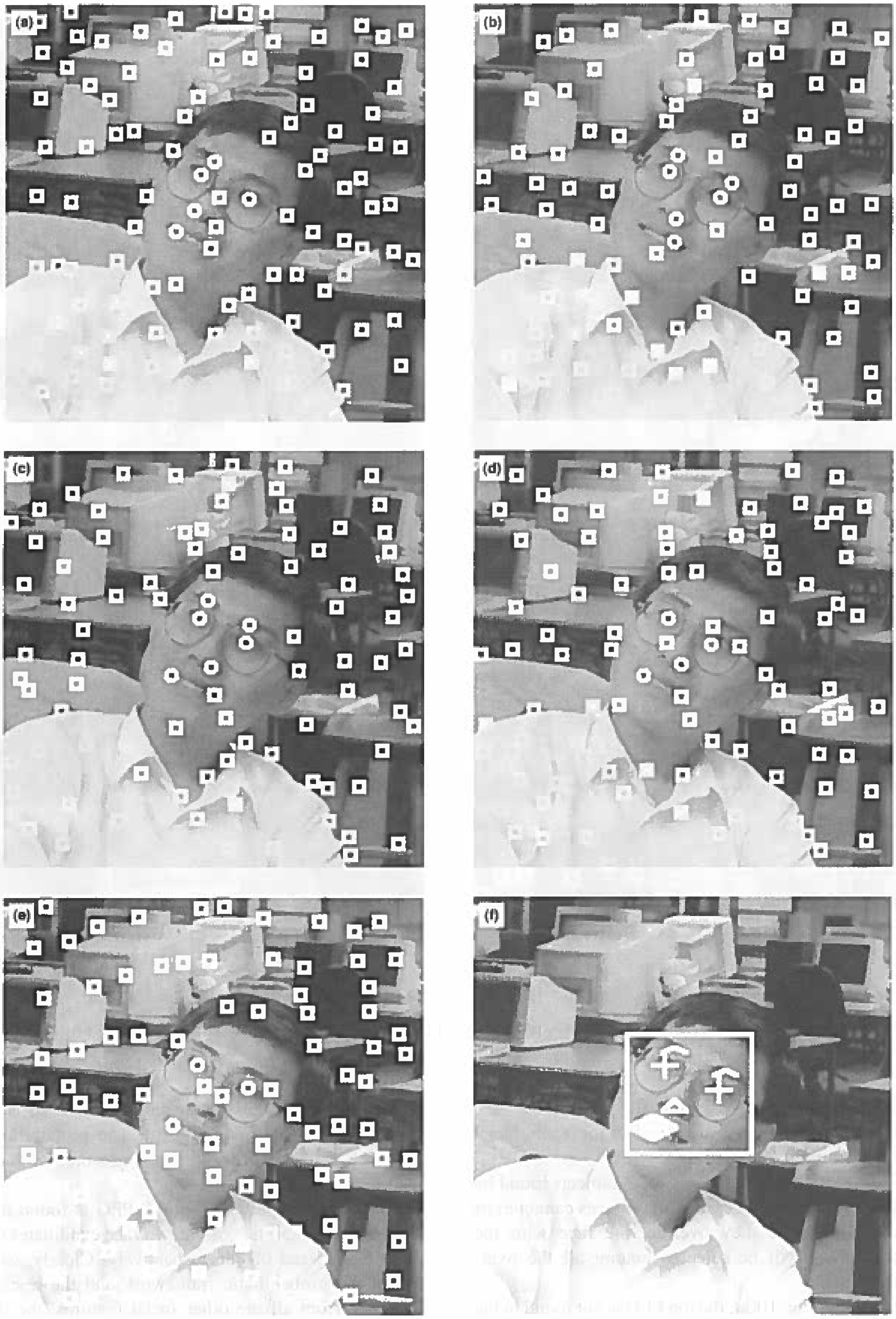


Fig. 16. Varying the orientation of the preattentive filter. The feature candidates that correspond to the true facial features are drawn as a circle instead of a square. (a) Orientation = − 60° (5 features detected). (b) Orientation = − 30° (6 features detected). (c) Orientation = 0° (6 features detected). (d) Orientation = 30° (4 features detected). (e) Orientation = 60° (3 features detected). (f) Face detected in the case of (b).

7.4. Preliminary results

We run our preliminary implementation of the algorithm on the test set of 60 images of size 256 × 256 pixels. These images contain faces at different scale but mainly in the fronto-parallel view with vertical orientation. Fifty-four are successfully detected, giving a 90% detection rate. Some of the successful results are shown in Fig. 11. We can see from the results that the algorithm is able to cope with variations in orientation and viewpoint (to a small extent), although we have made the assumption that the orientation is vertical and the viewpoint fronto-parallel. The algorithm also seemed to be robust to distractions such as glasses, and is also able to manage under a small amount of occlusion and absence of facial features. The time taken to run the algorithm on these images with a user-specified scale are about 10 s each.

Some of the unsuccessful cases are shown in Fig. 12. In the first image, the subject's eyebrows are actually very close to the eyes, and at that viewpoint, it is indistinguishable from the eyes. However, the algorithm groups two points in the hair region into a Hpair1 component group, forming false evidence leading to a wrong identification of facial features.

In the second image, the subject's left eyebrow (the right eyebrow in the image) coincides nicely with a dark horizontal strip in the background. As a result, the eyebrow is grouped with the background feature into a long feature, failing to be classified as a facial feature. A similar case happens to the mouth, which is also grouped with the strong edge caused by the shadow on the cheek, leading to a shift in the detected location of the mouth. Furthermore, due to the shift of the mouth location, the subject's right eye is not detected correctly because the actual location will give an incorrect geometric configuration of the face. Hence, a neighbouring point which has the next highest probability is selected instead.

In the third image, the face has rotated beyond the angle that the algorithm can cope. And since no other possible face candidates can be formed, no faces are detected. The fourth image shows a case of a false detection. The 'cheek' region of this face configuration is relatively free of features and hence the system is led to believe that this is a true instance of a face. In the 60 test images, 7 false detects were made, giving a false detection rate of 12%.

8. Approaches with invariance to scale, orientation and viewpoint

Many present approaches to solve the face detection problem have some invariance to scale, orientation and viewpoint changes, though either one or two of these are usually assumed to be fixed. We will examine how these approaches cope with scale, orientation and viewpoint changes.

The common approach to deal with scale variations is by examining the image at different scales and finding a match to a face template at each scale. Fixed-shape regions at different scales from the image are extracted, subsampled to the size of the template or filter, and then matched to the template. Experiments using this approach has been carried out by Lanitis et al. [13], Sung and Poggio [28], Yang and Huang [33]. The common problem with this approach is that the face template (or filter) is restricted to detecting only a single view and orientation of the face.

The feature-based approach used by Leung et al. [14], Yow and Cipolla [35] addresses the problem of orientation invariance by using the inter-feature distance or the affine geometry between the facial features. However, the facial features need to be extracted using a family of oriented filters – a rather computationally expensive task.

Chen et al. [3] extended the matching to 3 views (1 fronto-parallel and 2 profile views) using a fuzzy pattern matcher based on colour. Little geometric information is used. Sumi and Ohta [27] also attempted detection of profile views but their approach is largely based on image correlation.

9. Scale invariance

In this section, we will look at the effects of varying scale on the detection of faces. In our approach, two types of filters are used, the preattentive filter and the edge detection filter. Both of them are Gaussian derivative filters: the preattentive filter is a second derivative Gaussian while the edge detection filter is a first derivative Gaussian.

Fig. 13 shows the result of varying the scale of the preattentive filter from $\sigma = 3.0$ to $\sigma = 1.0$ while keeping the scale of the edge detection filter fixed at $\sigma = 3.0$. We observe that although 3 different values of σ are used, the face detected for all the 3 cases is the same. We also observe that at a scale ($\sigma = 1.0$) smaller than the one required for matched filtering ($\sigma = 3.0$), the correct facial features are still detected, though there is a much larger number of false detected feature points.

We subsequently vary the size of the image while keeping the scale of the preattentive filter constant at $\sigma = 1.0$. The scale of the edge detection filter is also reduced and kept at $\sigma = 1.0$. Fig. 14 shows the result for the image being reduced to 80%, 60%, 40% and 20% of the original size. We fail to detect the facial features when the face is too small because the image structure of these facial features are corrupted by quantization noise. However, we are successful in detecting the facial features of large faces even though our preattentive filter is small. This is because the size of the facial features are actually determined by the edge detection and edge linking process, and not by the scale of the preattentive filter.

We do a further test by varying the aspect ratio of the preattentive filter. Fig. 15 shows the result of varying

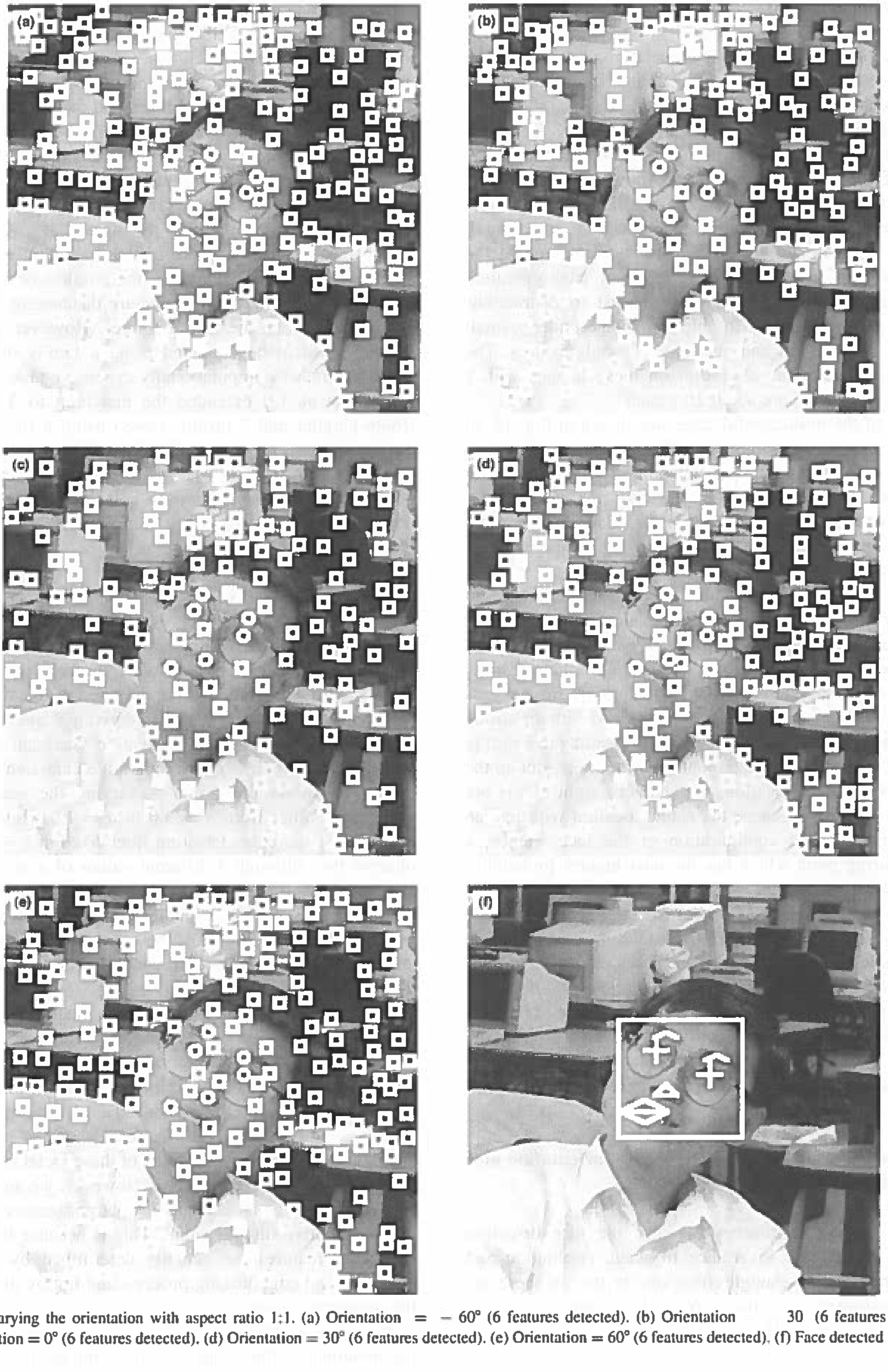


Fig. 17. Varying the orientation with aspect ratio 1:1. (a) Orientation = − 60° (6 features detected). (b) Orientation 30 (6 features detected). (c) Orientation = 0° (6 features detected). (d) Orientation = 30° (6 features detected). (e) Orientation = 60° (6 features detected). (f) Face detected in the case of (b).

the aspect ratio from 3:1 to 1:1. We observe that the facial features are still detected even though we reduce the aspect ratio to 1:1. The significance of this is that we can steer a 1:1 second derivative Gaussian exactly by using only 3 basis filters (Freeman and Adelson [9]), instead of using 16 basis filters to give a 1% error approximation for a 3:1 filter (Perona [21]) – a huge saving in computational requirements.

The Gaussian functional minimizes the product of localization in space and frequency (Marr and Hildreth [16]), but its trade-off between the signal-to-noise ratio and the accuracy of localization is well studied (Canny [2]). Since the edge detection filter is a first derivative Gaussian, choosing a small σ will result in noisy edges that are difficult to link. A large σ, however, will generate too much smoothing and may blur the image features, or even cause two separate edges to be smoothed into one. For an application of detecting the face of a person sitting in front of a computer terminal, a $\sigma = 1.0$ is found to be sufficient.

10. Orientation invariance

We will now look at the effects of varying the orientation. We use an image in which the subject's head is rotated approximately 30° to the right, i.e. at an orientation of −30° from vertical. We keep the preattentive filter at the 3:1 aspect ratio and rotate the filter from − 60° to 60° in 30° increments.

The results in Fig. 16 show that though the correct orientation is −30°, the facial features can still be detected by the filter at orientations of −60° and 0°. Thus, the algorithm can tolerate an orientation variation of about 30°.

We again reduce the aspect ratio of the preattentive filter. Fig. 17 shows the result of varying the orientation at an aspect ratio of 1:1. We observe that the facial features are now detected in all the different orientations of the filter. The significance of this is that we can make do without steerable filters completely. We can simply use only one single orientation of the preattentive filter and just examine the vicinity of the attention points for pairs of edges that are roughly parallel and have the correct polarity.

11. Viewpoint invariance

In Yow and Cipolla [35] we have shown that the Gaussian derivative filter (the preattentive filter described in this paper) is able to detect facial features under different viewpoint, even under profile view.

Fig. 18 shows the features detected by the preattentive filter in profile views of faces. The scale of the preattentive filter used is $\sigma = 3.0$ and the aspect ratio is 1:1. We observe that all the facial features which can be seen in the image are detected by the preattentive filter.

However, the difficulty in detecting faces under such viewpoints is that the facial features which we have chosen in our model can all be seen from only a limited range of viewpoints (mainly fronto-parallel). Thus, for general viewpoints (especially profile view), some of these features may be occluded and the evidential support for the face becomes low.

To overcome this, we look for additional features when we have a face hypothesis at a different viewpoint (e.g. profile view). We observe that in a profile view, there is a large region of the cheek that is roughly featureless. Hence we can mark out additional regions in the image that are the likely location of the cheeks, and examine the number, strength and orientation of edges in it. Face candidates that are formed from a single partial face group are examined for these cheek regions. The same Class space – Mahalanobis distance method is used to verify the group of features and regions as a valid profile view of a face.

The same situation applies to views which some of the facial features are occluded. The different cheek regions that are used for the different views are shown in Fig. 19.

12. Results

12.1. Detection under scale, orientation and viewpoint variations

Our system is retrained on a different set of images, which are obtained from different sources such as CCD camera output, as well as scanned photographs. The number of faces in these images totalled about 100.

To test the algorithm against scale, orientation and viewpoint variation, we get subjects to sit in front of a workstation mounted with a Pulnix monochrome CCD camera. The subject is free to turn his head or move his chair forward. A total of 9 images are taken from 10 subjects at varying distances and viewpoints (5 fronto-parallel view, 2 with head rotated to either side, and 2 profile views). In addition, a further 10 images of 2 subjects in the same image is taken, totalling the number of faces in the test to 110. The images used are 256 × 256 pixel resolution.

We implement the face detection algorithm as described in the preliminary implementation but we do away with the user-specification of the filter scale. We use only one single scale and orientation of the filter for the preattentive feature selection process. The scale of the preattentive filter is chosen to be the same as the edge detection filter ($\sigma = 1.0$) so that we only need to smooth the image once. However, the time taken to process the images are increased to about 90 s due to the much larger number of feature points detected. The intermediate results are shown in Fig. 20.

Comparing the results of Fig. 20 with that of Figs 8 and 9, we observe a large increase in the number of feature points (466/81 ≈ 5.75 times increase in the number of points). However, the perceptual grouping process is able to reduce the number of face candidates down to only two. The belief

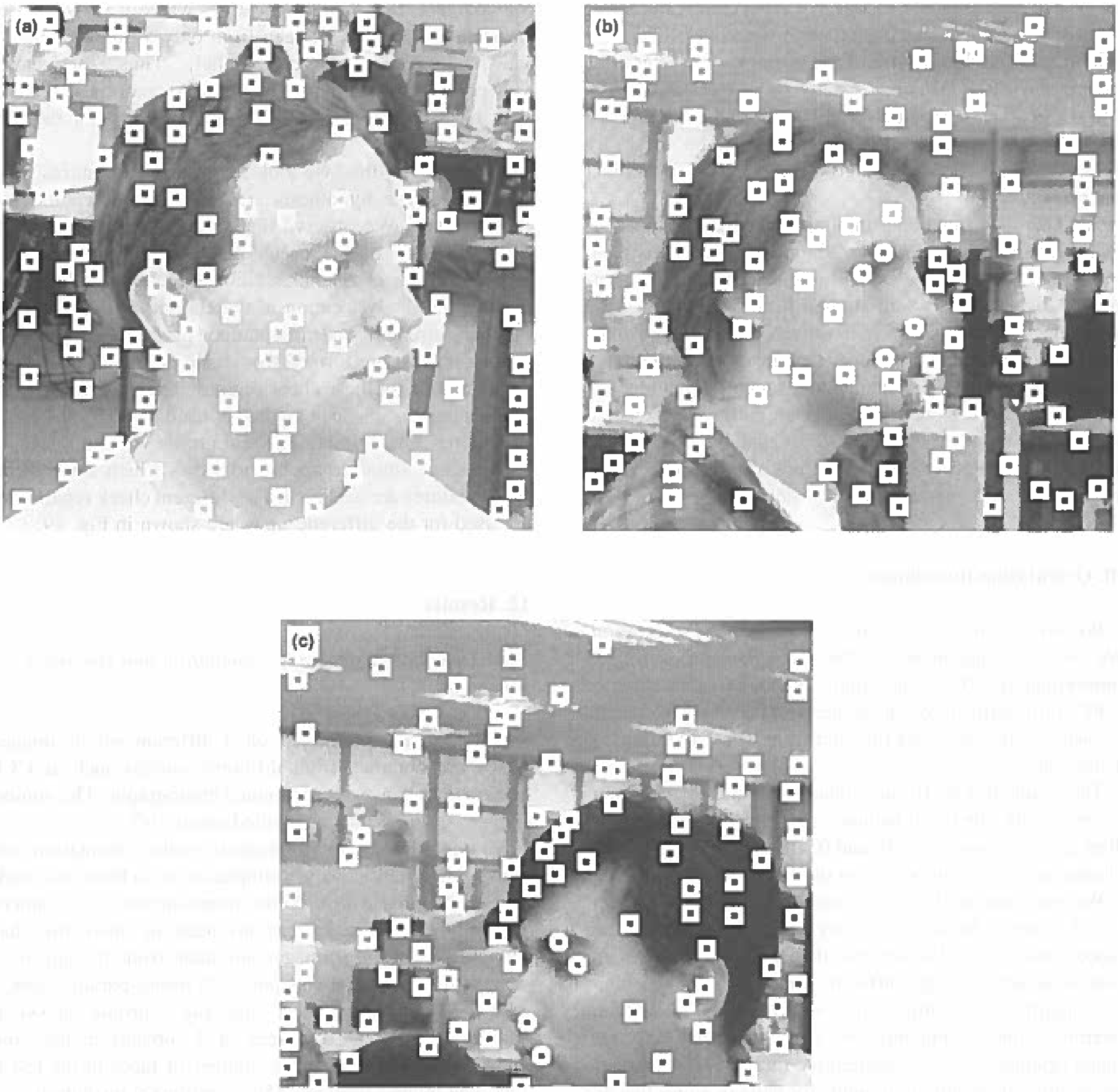


Fig. 18. Detecting features in profile views. (a) 121 points (4 features detected). (b) 122 points (4 features detected). (c) 113 points (4 features detected).

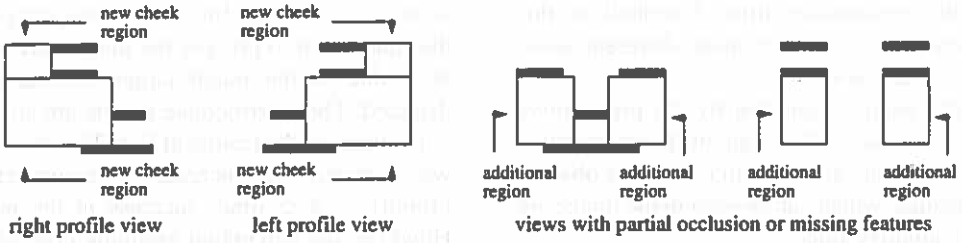


Fig. 19. The additional cheek regions used under different viewpoints and when facial features are missing or occluded.

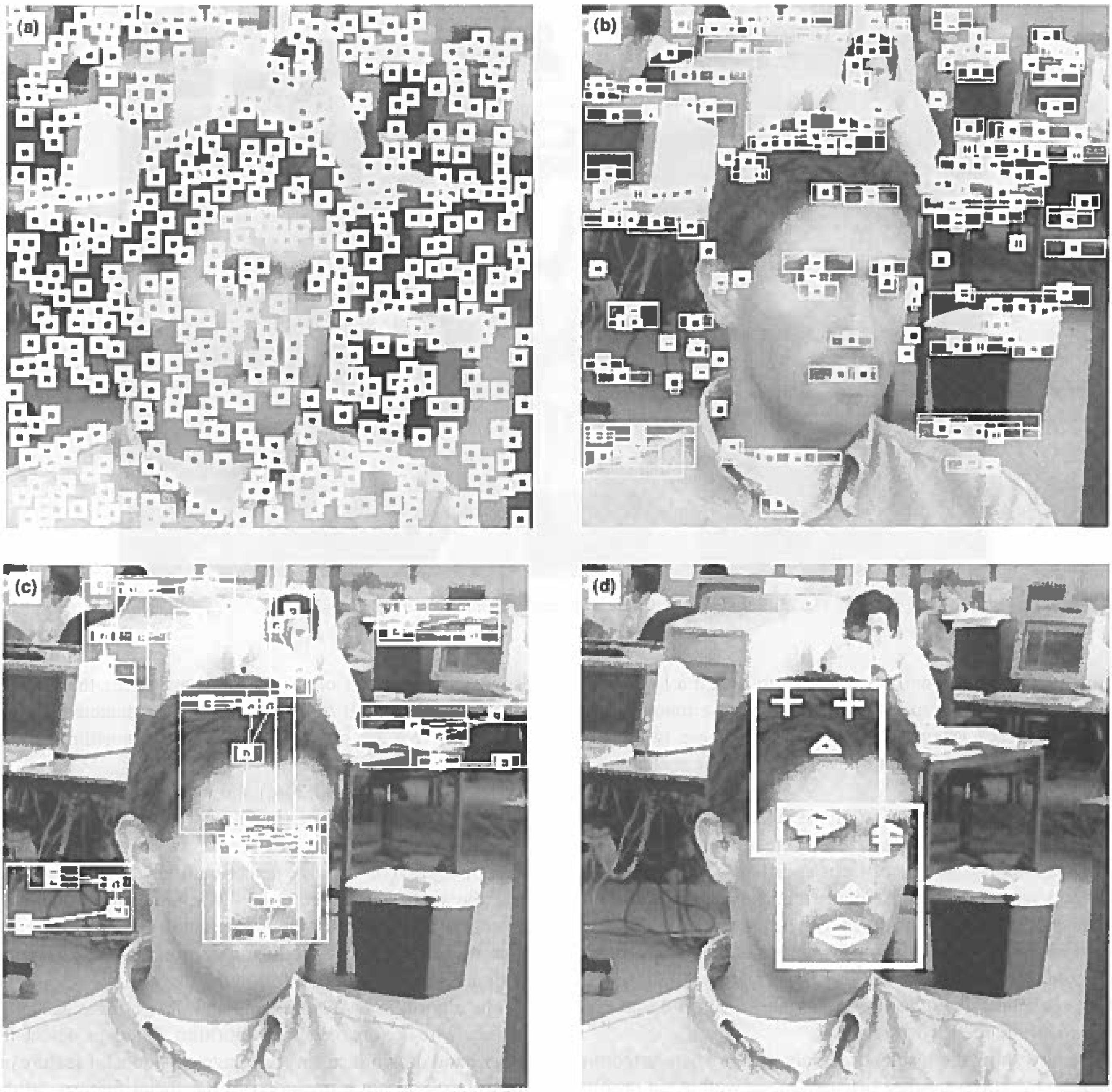


Fig. 20. (a) Interest points (466 points). (b) Feature regions (180 points). (c) Partial Face Groups (28 top, 10 bottom, 5 left, 1 right). (d) Face candidates (2 faces). The probabilities associated with the upper and lower face candidates are 0.6124 and 0.9578 respectively.

Table 1
Summary of results

Views	Number of faces	True positives	False positives
Fronto-parallel	70	65 (92.9%)	15 (21.4%)
Fronto-rotated	20	16 (80.0%)	7 (35.0%)
Profile	20	12 (60.0%)	9 (45.0%)
Total	110	93 (84.5%)	31 (28.1%)

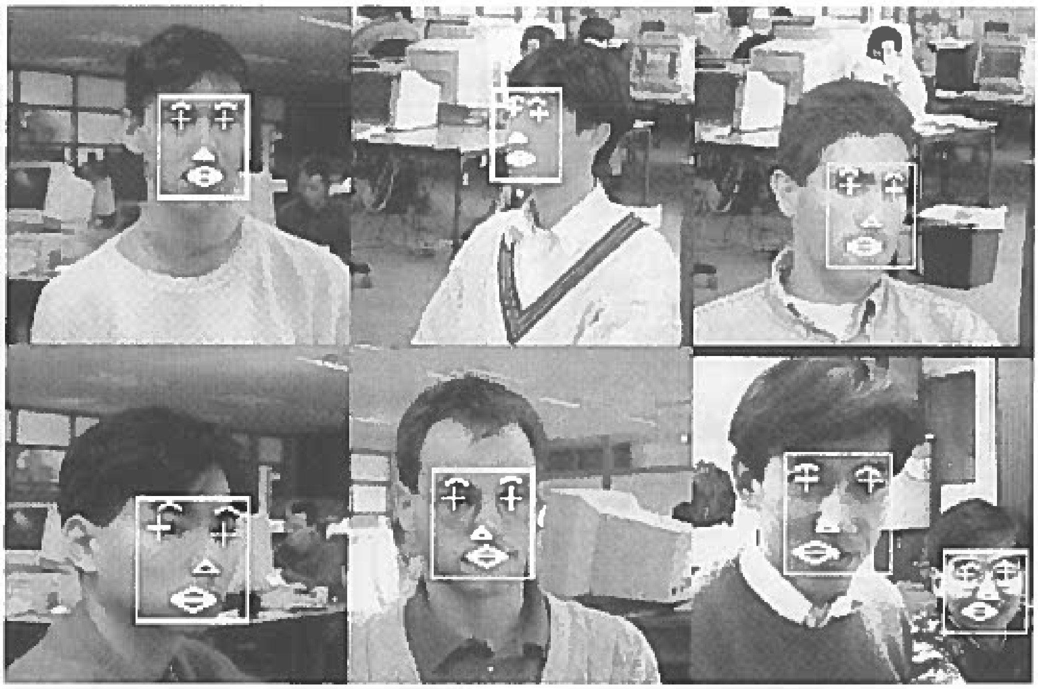


Fig. 21. Result of face detection on various face images at different scales.

propagation process will also be able to assign a high confidence value to the true positive face because many of the component face groups are present. Since these two faces overlap, only one face (the one with the higher probability) will be selected.

Table 1 gives a summary of the results of the test. We achieved an overall successful face detection rate of 85% on a database of 110 images of faces at different scale, orientation and viewpoint. A false detection rate of 28% was also obtained. Some of the results of the test are shown in Fig. 21. We observe that the algorithm is able to handle a good range of scale variations. Due to the larger number of feature points detected, the algorithm now takes about 90 s to run on a SUN Sparc20 workstation.

We also show the results of the algorithm when tested on images with different face orientation, as well as on profile views of the face. The results in Figs 22 and 23 show that the algorithm is able to cope with variations in orientation and viewpoint.

12.2. Evaluation of the algorithm on the CMU-database

We evaluate the performance of the algorithm by testing it on the face database used by Rowley et al. [22]. These images can be obtained from: `Http://www.ius.cs.cmu.edu/IUS/dylan_usr1/har/usr0/har/faces/test/`. Fig. 24(a)–(f) show the performance of our algorithm on some of the images in the database.

One shortcoming of our algorithm is that it is unable to detect faces which are too small. In the database used by Rowley et al., most of the images have faces that are too small (less than 60 × 60 pixels) to be detected by our algorithm. We are only able to test our algorithm on less than 20% of the images.

Fig. 24(a), 24(b) and 24(c) show successful detection of all the faces in the image. In Fig. 24(d) we failed to detect the largest face in the image because we only use one single scale ($\sigma = 1.0$) of the edge detection filter. The face is large and the image is rather noisy. The small σ of the edge detection filter is insufficient to smooth out the noise and thus we are unable to produce a continuous edge segment with a standard boundary following algorithm.

The algorithm also works on hand-drawn faces as shown in Fig. 24(e). However, the algorithm failed to detect the other hand drawn faces in the image as the facial features in these faces do not correspond to the facial features in our model. False detects can also be seen in Fig. 24(f). Some incorrect facial features turned out to have higher probabilities than the true facial features and as a result, the true face is displaced by the slightly incorrect face.

13. Discussion

Though our algorithm is invariant to scale to some extent, it is by no means universal. It is nevertheless not sufficient for cases where the face is very small, or when the scale of the spatial filters are very large compared to the features. Feature points will move or even disappear under scale-space filtering (Witkin [32]), and facial features will take on a very different image structure due to quantization

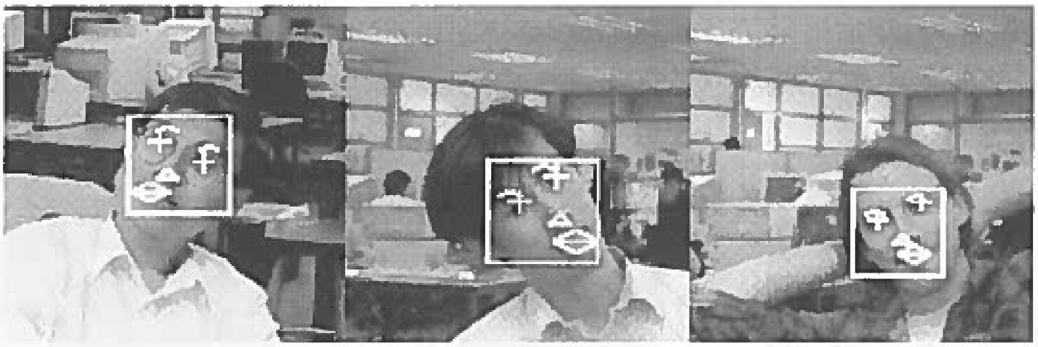


Fig. 22. Result of face detection on face images at different orientation.

noise. Hence, the evidence that can be extracted from the image will be quite different at different scales. So a different set of facial features, or even head or body features (e.g. face texture, hair texture, head and shoulder boundaries, silhouettes of the body) need to be used at different scales. However, the proposed perceptual grouping and probabilistic framework will remain the same throughout the different scales.

The difficulty with feature-based algorithms is that the image features can be badly corrupted due to illumination, noise or occlusion. Feature boundaries can be weakened by illumination, and shadows can cause numerous strong edges that render perceptual grouping algorithms useless. Worse still, such failures usually occur in the early, low-level stages. A truly robust system must make use of a huge number of features, each of which must be invariant to different kinds of imaging conditions.

Another minor drawback of this approach is that the model specifications (i.e. specifying what the features are, which spatial relationship is important, etc.) are done by hand. It will be desirable to automatically learn which features are important so that it can be extensible to detect objects in other domains (e.g. cars). No successful method of doing this is known at the moment, and investigation is under way.

Our future work will be aimed at coping with smaller scales of faces. Approaches used will be finding the head boundary, which is a difficult task because the boundary may not be clearly defined when background clutter is present. Textual algorithms will also be investigated to harness the texture of hair and skin in detecting faces.

14. Conclusion

We have proposed a feature-based face detection framework which extracts interest points using spatial filtering techniques, groups these points into face candidates using perceptual grouping principles, and selects true candidates from false ones using a probabilistic framework. We also make use of model knowledge as evidence to improve the confidence of faces in the image. As a result, we can make less assumptions about the image structure of faces and thus make our algorithm more robust to different imaging conditions. The algorithm is shown to be able to be easily extended to work for different scale, orientation and viewpoint of the face. The framework can be further extended to more difficult imaging conditions by adding more components to the face model and finding more evidence in the image to support the face hypotheses.

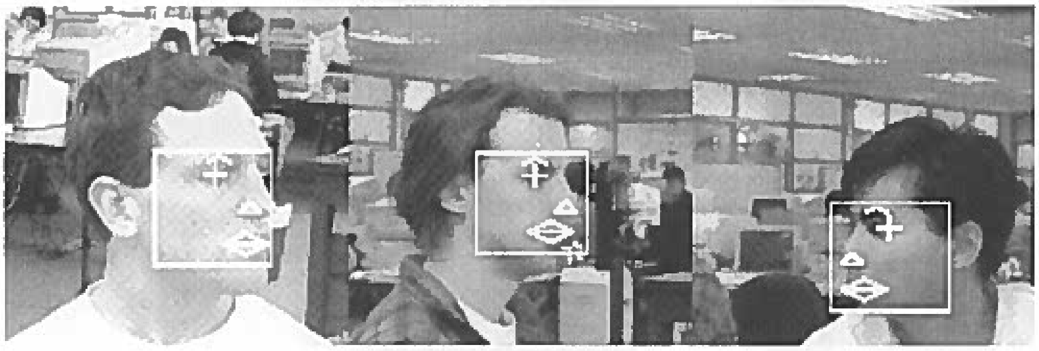


Fig. 23. Result of face detection on face images at profile views.

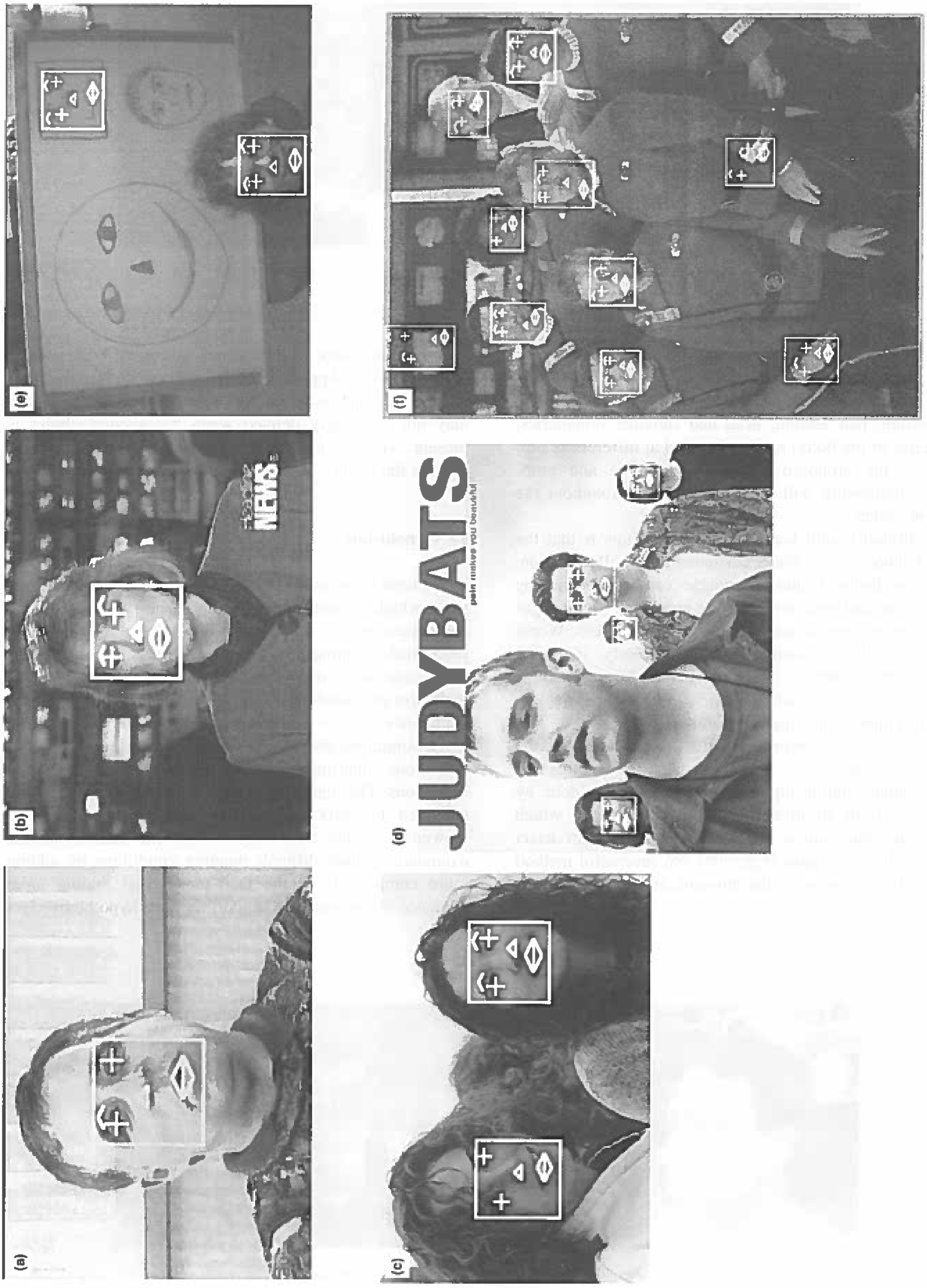


Fig. 24. Result of face detection on CMU face database images. The size of the images are (a) 320 × 240 pixels; (b) 320 × 240 pixels; (c) 414 × 273 pixels; (d) 716 × 684 pixels; (e) 640 × 480 pixels; (f) 662 × 874 pixels.

References

[1] C.H. Ballard, C.M. Brown, Computer Vision, Prentice-Hall, 1982.

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

[3] Q. Chen, H. Wu, M. Yachida, Face detection by fuzzy pattern matching, in: Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, 1995, pp. 591–596.

[4] G. Chow, X. Li, Towards a system for automatic facial feature detection, Pattern Recognition 26 (12) (1993) 1739–1755.

[5] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models – Their training and application, Compuer Vision and Image Understanding 61 (1) (1995) 38–59.

[6] I. Craw, D. Tock, A. Bennett, Finding face features, in: G. Sandini (Ed.), Proc. 2nd European Conf. on Computer Vision, Italy, 1992, Springer-Verlag, pp. 92–96.

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

[8] Y. Dai, Y. Nakano, H. Miyao, Extraction of facial images from a complex background using SGLD matrices, in: Proc. Int. Conf. on Pattern Recognition, vol. A, Jerusalem, 1994, IEEE Computer Society Press, pp. 137–141.

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

[10] R.M. Haralick, Texture features for image classification, IEEE Trans. Systems, Man, and Cybernetics 3 (6) (1973) 610–621.

[11] K. Koffka, Principles of Gestalt Psychology, Harcourt, Brace and World, 1935.

[12] W. Kohler, Gestalt Psychology, Liveright, New York, 1947.

[13] A. Lanitis, C.J. Taylor, T.F. Cootes, An automatic face identification system using flexible appearance models, Image and Vision Computing 13 (5) (1995) 393–401.

[14] T. Leung, M. Burl, P. Perona, Finding faces in cluttered scenes using labelled random graph matching, in: Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, 1995, pp. 637–644.

[15] D.G. Lowe, Perceptual Organization and Visual Recognition, Kluwer, Boston, 1985.

[16] D. Marr, E.C. Hildreth, Theory of edge detection, Proc. Royal Soc. London B207 (1980) 187–217.

[17] R. Mohan, R. Nevatia, Perceptual organization for scene segmentation and description, IEEE Trans. Patt. Anal. Machine Intell. 14 (6) (1992) 616–634.

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

[19] S.E. Palmer, The psychology of perceptual organization: A transformational approach, in: J. Beck, B. Hope, A. Rosenfeld (Eds.), Human and Machine Vision, Academic Press, New York, 1983, pp. 269–339.

[20] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufman, 1988.

[21] P. Perona, Steerable-scalable kernels for edge detection and junction analysis, in: G. Sandini (Ed.), Proc. 2nd European Conf. on Computer Vision, Italy, 1992, Springer-Verlag, pp. 3–18.

[22] H.A. Rowley, S. Baluja, T. Kanade, Human face detection in visual scenes, Technical Report CMU-CS-95-158, CMU, July 1995.

[23] S. Russell, J. Binder, D. Koller, K. Kanazawa, Local learning in probabilistic networks with hidden variables, in: Proc. Int. Joint Conf. on Artificial Intelligence, 1995.

[24] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1994.

[25] S. Sarkar, K.L. Boyer, Integration, inference, management of spatial information using bayesian networks: Perceptual organization, IEEE Trans. Patt. Anal. Machine Intell. 15 (3) (1993) 256–273.

[26] B. Schiele, A. Waibel, Gaze tracking based on face-color, in: Proc. Int. Workshop on Automatic Face and Gesture Recognition, Zurich, 1995, pp. 344–349.

[27] Y. Sumi, Y. Ohta, Detection of face orientation and facial components using distributed appearance modeling, in: Proc. Int. Workshop on Automatic Face and Gesture Recognition, Zurich, 1995, pp. 254–259.

[28] K.K. Sung, T. Poggio, Example-based learning for view-based human face detection, Technical Report A.I. Memo 1521, CBLC Paper 112, MIT, 1994.

[29] T.I.P. Trew, R.D. Gallery, D. Thanassas, E. Badiqué, Automatic face location to enhance videophone picture quality, in: Proc. 4th Brit. Machine Vision Conf., Springer-Verlag, 1993, pp. 488–497.

[30] A. Triesman, Perceptual grouping and attention in visual search for features and objects, J. Experimental Psychology: Human Perception and Performance 8 (2) (1982) 194–214.

[31] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognitive Neuroscience 3 (1) (1991) 71–86.

[32] A. Witkin, Scale space filtering, in: A. Pentland (Ed.), From Pixels to Predicates, Ablex Publishing Company, New Jersey, 1986, pp. 5–19.

[33] G. Yang, T.S. Huang, Human face detection in a complex background, Pattern Recognition 27 (1) (1994) 53–63.

[34] K.C. Yow, R. Cipolla, Finding initial estimates of human face location, in: Proc. 2nd Asian Conf. on Computer Vision, vol. 3, Singapore, 1995, pp. 514–518.

[35] K.C. Yow, R. Cipolla, Towards an automatic human face localization system, in: Proc. 6th British Machine Vision Conf., vol. 2, 1995, pp. 701–710.

[36] A.L. Yuille, P.W. Hallinan, D.S. Cohen, Feature extraction from faces using deformable templates, Int. J. Computer Vision 8 (2) (1992) 99–111.