Original approach for the localisation of objects in images

R. Vaillant
C. Monrocq
Y. Le Cun

Indexing terms: Neural networks, Localisation, Image analysis, Television

Abstract: An original approach is presented for the localisation of objects in an image which approach is neuronal and has two steps. In the first step, a rough localisation is performed by presenting each pixel with its neighbourhood to a neural net which is able to indicate whether this pixel and its neighbourhood are the image of the search object. This first filter does not discriminate for position. From its result, areas which might contain an image of the object can be selected. In the second step, these areas are presented to another neural net which can determine the exact position of the object in each area. This algorithm is applied to the problem of localising faces in images.

1 Introduction

The detection and localisation of faces in an image has many applications in various domains: surveillance, television audience polling etc. We propose a new method for this task which

(i) does not require any hypothesis about the position of the face in the image, or on its scale;

(ii) does not require any hypothesis on the background; and

(iii) can be implemented to operate at a fraction of video rate (5–10 images/s) with current technology.

The main feature of the method is to train a neural network to detect the presence or absence of a face in its input window, and to scan this network over possible locations in the image. Because of the nature of the neural-network architecture we used, this process can be carried out very efficiently without requiring actually recomputation of the entire network state at each location. The scanning is performed on several versions of the image at various scales, resulting in an efficient, scale-independent detector and locator.

Several approaches to this problem have been proposed in the literature. There are two main classes of methods:

(*a*) The first kind of approach relies on the use of a synthetic model of a face. In Reference 2, the authors represent a face as a combination of two parallel lines, which are the sides of the face, and two arcs of a circle for the chin and the top of the face. Yuille *et al.* [10] suggest that each part of the face be represented as a deformable element which is searched for in the image by minimising an energy. Vincent *et al.* [9] locate these different parts using neural nets. Craw *et al.* [1] have a similar approach. This kind of technique has the following difficulties: the computation time for adjusting the model could be long and the choice of the initial position of the model is quite difficult.

(*b*) The second kind of approach relies on building a classifier which processes constant-size images, and indicates whether they correspond to a face or not. Turk [7, 8] uses principal-component analysis. Neural nets are used in Reference 5.

2 The database

To detect some specific elements in an image, it is necessary to describe the primitives that must be detected in a way which is compatible with the algorithm used. One of the main advantages of some advanced neural-net architectures is their ability to process raw (or almost raw) images. The problem of finding (and computing) the appropriate representation for the classifier is greatly eased. Our database is composed of many examples of small-size images of 'faces' and 'nonfaces'.

2.1 Formation of the database: image acquisition

28 volunteers of both sexes and various ages were asked to walk towards a camera, starting from a distance of 5 m from the camera, to a distance of about 3 m from the camera. The subjects were asked to talk, and change facial expression and head attitude, while walking. To make the problem simpler, we ask those subjects who wore glasses to remove them, since the glasses reflect light and can introduce highlights in the images. Because of the varying distance of the subjects from the camera, the observed faces had widely varying sizes (the ratio of the sizes between the different images of the sequence is 3 : 1). To take into account the variations in lighting conditions, we acquired two sequences of images: in the first, there was only one light behind the camera, while in the second, there was also more diffuse lighting. A supplementary sequence, without faces, was also acquired.

The images were smoothed with a zero-mean Laplacian filter. They were also normalised for the mean and the standard deviation. The mean of the pixels of each image was set to 0 and the standard deviation to 1.

Paper 1301K (E4, C4), first received 10th November 1993 and in revised form 25th April 1994

R. Vaillant and C. Monrocq are with the Laboratoire Central de Recherches, Thomson-CSF, Domaine de Corbeville, 91404 Orsay Cedex, France

Y. Le Cun is with Room 4G-332, AT&T Bell Laboratories, Crawford Corner Road, Holmdel, NJ 07733, USA

2.2 Formation of the database: extraction of patches

As briefly mentioned in the Introduction, the neural net is given a small window taken from the input image, and is asked to activate its output if a face is present in the window. The size of the window was chosen to be 20×20 pixels. We chose this size because it is close to the minimum resolution which allows unambiguous distinction between faces and nonfaces. In Reference 6, it is mentioned that a size 16×16 is the lower limit at which the human can detect a face.

To handle the scale variation, three approaches are possible. The first is to train the neural net to detect faces independent of their size in the window. The second to train a separate neural net for each range of size, and combine their outputs. The third approach, which is that we used, is to use a single neural network, and scan it over several versions of the input image at various resolutions. The outputs from the network at various scales are then combined.

To create examples of 20×20 pixel images of faces and nonfaces, we manually segmented the whole database by entering, for each image, the point $m_1 = (x_1, y_1)$ between the eyes, and the point $m_2 = (x_2, y_2)$ at the centre of the mouth. The second point gives information about the orientation and scale of the face. The area of the face in the original image was reduced to a patch of size 20×20 using appropriate scaling factors chosen from a discrete set of scaling factors. As is shown later, the same scaling factors were used when the algorithm was applied to the whole images. We used seven different scaling factors. Consequently, the faces did not always completely fill the patch of size 20×20. This patch was included in a bigger patch of size 48×32 (Fig. 1).

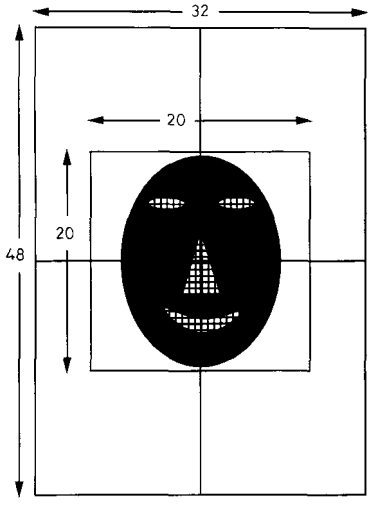

Fig. 1 *Geometry of the patches that have been extracted for the database*

The database contains 1792 patches with a face. We have formed an equal number of images without faces, which we will call background patches, using the sequence of 32 images.

3 The training

3.1 General principle

Several neural-net architectures were tried. The simplest one has no hidden layer, while the others have multiple convolutional hidden layers [4]. These networks were trained with the backpropagation algorithm taking into account the shared weights. The networks had the following points in common:

(i) an input layer of size 20×20; each of the neurons of this layer is fed with the corresponding pixel of the patch

(ii) an output layer which contains only one neuron; this neuron indicates if the presented patch corresponds to a face or not.

3.1.1 A neural net without hidden layer: The neural net does not include any hidden layer. We use it for analysing the complexity of our problem. It contains 401 weights.

3.1.2 A shared-weight neural net: This neural net comprises three hidden layers. Each of the hidden layers is divided into four small images (or feature maps). This net uses shared weights following the ideas described in References 3 and 4. Fig. 2 shows the architecture of the neural net. Each neuron of each map of the first hidden layer is connected to 5×5 neurons of the input layer. The weights are shared in the map. Each neuron of each map of the second hidden layer is connected to 2×2 neurons of the corresponding map of the first hidden layer. The weights are shared. The neuron of each map of the third hidden layer is connected to each neuron of the corresponding map of the second hidden layer. The neuron of the output layer is connected to the four neurons of the third hidden layer. This net has 1157 free parameters (but many more connections because of the weight sharing). There are many well known advantages in using shared-weight neural-net architecture (fewer free parameters, better generalisation, distortion invariance). In our context, shared-weight architectures have another decisive advantage. For our application, the network must be replicated (or scanned) over a large image (say 256×256 pixels). Now, since each layer of the network essentially performs a convolution (with a small-size kernel), a large part of the computation is in common between networks applied at two neighbouring locations. This redundancy can be eliminated by performing the convolution corresponding to each layer on the entire image at the same time. The overall computation amounts to a succession of convolutions and nonlinear transformations over the entire image.

3.2 What must the neural net learn?

Once the architecture of the net is chosen, we need to know what the neural net must learn. We propose three areas:

(*a*) Training with the goal of performing a complete localisation the elements of the database are presented to the neural net. If the patch presented corresponds to a perfectly centred face, the desired output is α, else the desired output is $-\alpha$. This is the natural choice which makes a direct use of the two classes of our problem.

(*b*) Training with the goal of performing a rough localisation. The elements of the database are presented to the neural net as a perfectly centred patch or a shifted patch. This means that we feed the input layer of the neural net with an image extracted from the patch 48×32 whose size is 20×20 and whose centre is placed at (x, y) pixels from the centre of the patch of the database. If the image is perfectly centred, the desired output is α. If the image is shifted, the desired output is $\alpha[2 \exp\{-\lambda\sqrt{(x^2 + y^2)} - 1\}]$. The desired output is an exponentially decreasing

function of the shift. If the image presented is a background, the desired output is $-\alpha$. Our goal is to train the net to give a medium answer when it encounters a shifted face and to give a maximum answer when it encounters a perfectly centred face. So when the neural net will be applied to a complete image, the answer obtained will be smooth all around the face. The areas of the image which correspond to face will be easy to detect. The drawback is that the position of the centre of the face will not be very precise.

(*c*) Training with the goal of performing a precise localisation. The elements of the database are presented perfectly centred with a desired output α or shifted with a desired output $-\alpha$. The background images are not presented. This neural net must be able to localise the centre of the face precisely if the input layer is fed with faces more or less centred.

We can also notice that the two last training techniques multiply the number of patterns in the database. Indeed, $336 = (48 - 20) \times (32 - 20)$ different images are formed from each original image of the database. Even if these images are not completely independent patterns, we can assume that the generalisation rate is correctly estimated and we do not have overfitting.

3.3 Results of the training

In a first step, we have tested the two neural nets which are described in Section 3.1 and use the first two learning methods which are presented in Section 3.2. The learning set is formed with half the database and the test set is formed with other part of the database. With a classical workstation (Sun4 SPARC), several hours are needed for some of the training sessions.

3.3.1 Training for a complete localisation: Fig. 4 shows the evolution of the quadratic error and the rate of well classified example. The quadratic error is defined as $(1/2N) \sum_{n=1}^{N} (d^n - o^n)^2$. o^n is the output obtained when the example n is presented and d^n is the desired output. An example is assumed to be well classified if o^n and d^n have the same sign. These values are measured after a presentation of the whole training base to the neural net.

First, we can note that the two neural nets have results which are roughly equivalent. On the test set, the quadratic error decreases quickly and the rate of correct recognition increases towards 96%.

These results could appear to be satisfactory. In fact, they are not. Indeed, we plan to segment an image in areas which correspond to a face and in areas which do not correspond to a face. We apply the neural net with a shared weight to a standard image whose size is 256×256. The output will be an image of size $126 \times 126 = 15876$. The size of the output image is different from the input as there is one hidden layer which subsamples its input, thereby dividing the size of the input by two. If the rate of images which are well classified is 96%, the image will include 635 positive answers which will probably correspond to false alarms. This result cannot be exploited: there are too many false alarms.

Fig. 3*b* shows the result obtained when the shared weight net is applied to the image of the Fig. 3*a*. The grey level of the pixels is proportional to the answer of the neural net. This image is scaled so that its resolution is 86×86. There are 95 pixels with a positive answer (5.6% of the points of the output image).

3.3.2 Training for a rough localisation: Fig. 5 shows the quadratic error and the rate of well classified examples for a rough localisation. We can note the following:

(i) The first net does not succeed in learning. The quadratic error and the rate of well classified examples stops changing significantly after a few iterations.

(ii) For the second net, the error and the recognition rate decrease more slowly than for complete localisation.

(iii) There is no overfitting. The generalisation rate does not decrease at the end of the training. For training for the complete localisation, they decrease at the end of the training.

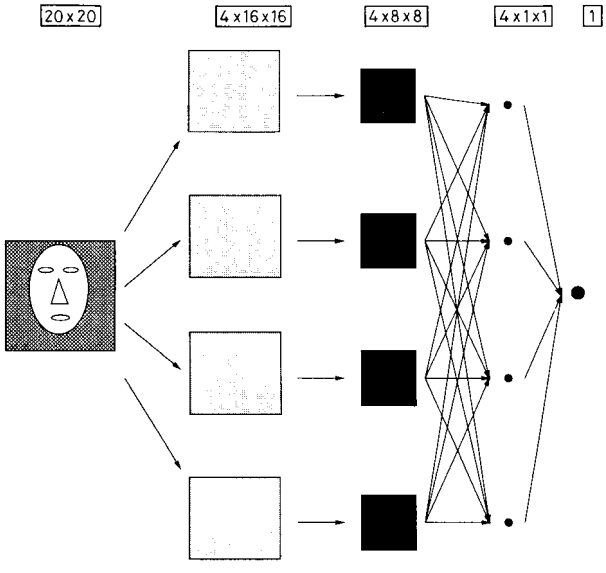

Fig. 2 *Architecture of the neural net*

This indicates that this problem is more difficult than that previously discussed.

The generalisation rate obtained at the end of the training phase is lower than the rate we obtained for the complete localisation. Consequently, when we apply our net to a complete image, it will produce a greater number of false alarms. On the other hand, the false alarms can easily be separated from the correct alarms. Indeed, when a face is present in the image, there is a complete area where the neural net gives a positive answer.

Fig. 3*c* shows the result obtained when this net is applied to the image of Fig. 3*a*. There are 181 pixels with a positive answer. It is equivalent to an estimated generalisation rate of 90%. It is important to note the distribution of positive answers. They are grouped in about ten areas. Each of these areas could be considered as an hypothesis for the detection of a face and could be subject to further processing, as explained below.

3.4 Training for a precise localisation

Fig. 3*d* shows the results obtained when the shared weight net is applied to the whole image. There are 79 pixels with positive answers, scattered over the whole image. This is normal as the net has not been trained to give identical answers for a pixel and its neighbours.

4 Application to images

We wish to obtain an algorithm for the localisation of faces in images which does not make any hypothesis

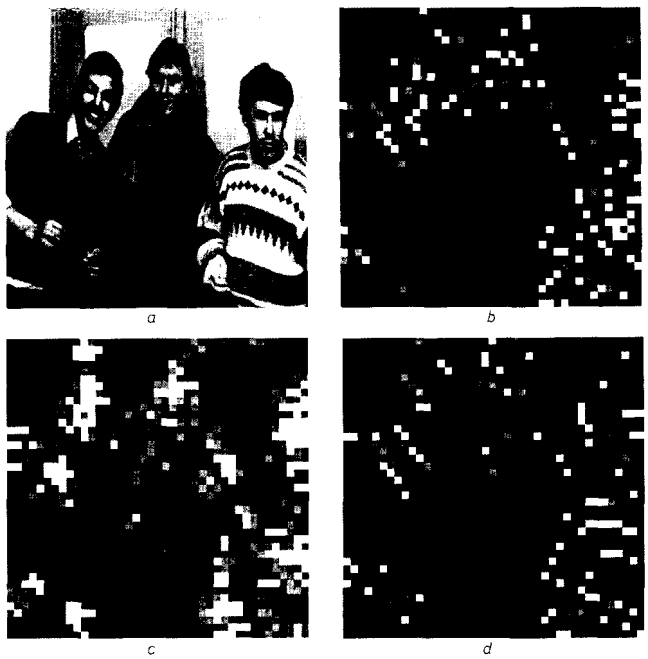

Fig. 3 *Neural nets applied to an image*

a The image
b The neural net with shared weight trained for a complete localisation
c The neural net with shared weight trained for a rough localisation
d The neural net with shared weight trained for a precise localisation

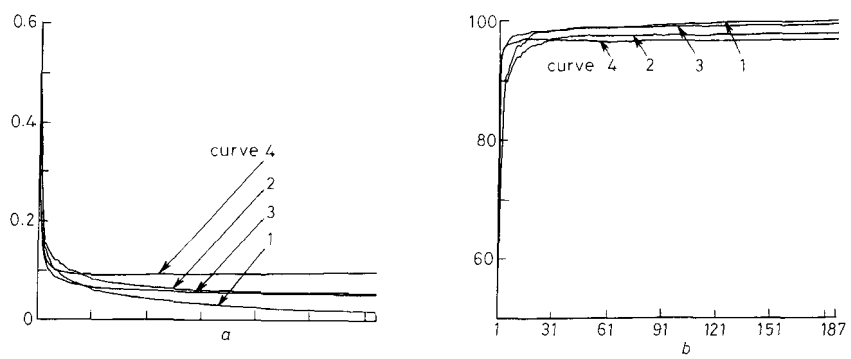

Fig. 4 *Training for a complete localisation*

a Quadratic error
b Rate of good classified
Curve 1: training with the shared-weight neural net
Curve 2: generalisation with the shared-weight neural net
Curve 3: training with the completely connected neural net
Curve 4: generalisation with the completely connected neural net

about the scale of the face in the image. The system described in the previous Sections requires that the face is observed at a fixed size. We apply these results to an image where the size of observed faces is unknown by processing this image at several resolutions with the same network. The complete algorithm is as follows:

(i) Several versions of the original image are created at different scales (the set of scaling factors is determined in advance). The shared-weight neural nets trained for rough localisation are scanned over each of the images. Fig. 6 shows the output of the net for each scale.

(ii) We look for 'blobs' of positive values in the output maps produced by the network. Each of the blobs is considered as a good candidate (an hypothesis) for fine detection of faces (see Fig. 7).

(iii) We apply the neural net trained for a precise localisation to each hypothesis and search for that which gives the maximal answer. If it is larger than some threshold (at this stage, some of the hypothesese may be removed), the hypothesis is assumed to be valid and the point with the maximum answer is taken as the centre of the face.

(iv) The different valid hypothesese which correspond to a single face are grouped. Indeed, a single face can be detected at two different scales. This is quite frequent, as the resolutions used are not very different, and the faces in the database are not very precisely normalised to the same scale. To group the different valid hypothesese, we consider the area which they describe in the original image. If two hypothesese are conflicting, i.e. their corresponding areas intersect, we retain only that corresponding to the highest answer. Fig. 8 shows the set of retained hypothesese. A rectangle is drawn, in the initial image, around the area associated with each hypothesis. The size of the rectangle is computed from the resolution at which the hypothesis was formed.

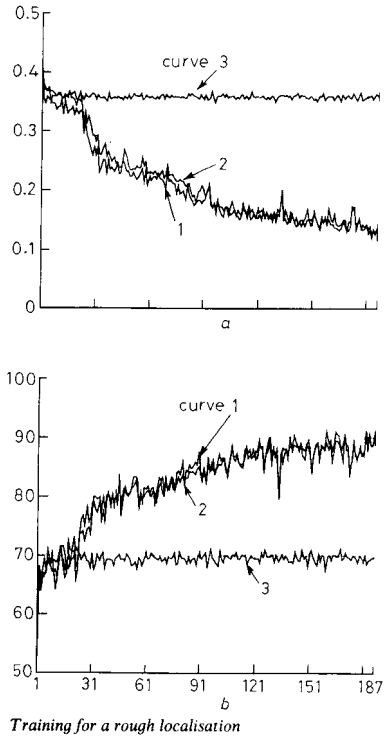

Fig. 5 *Training for a rough localisation*

a Quadratic error
b Rate of good classified
Curve 1: training with the shared-weight neural net
Curve 2: generalisation with the shared-weight neural net
Curve 3: training with the completely connected neural net

5 Conclusion

We have presented an algorithm for the detection of faces in images using shared-weight replicated neural networks. In a first step, a first neural net forms rough hypotheses about the position of faces. These hypotheses are then verified in the second step using a second neural network. We have also shown that the algorithm applies to images where the size of the face is unknown *a priori*.

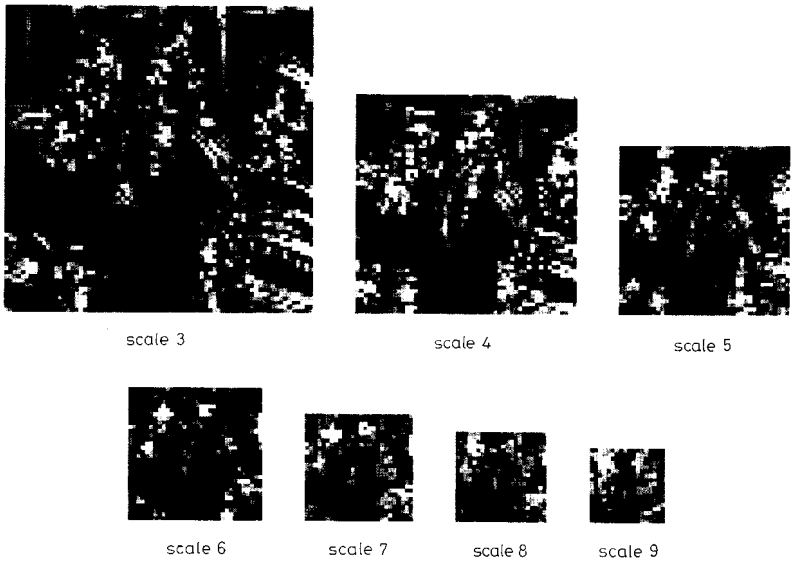

Fig. 6 *Images resulting from the shared-weight neural net trained for a rough localisation*

The input image is treated with several resolutions. Hypotheses are formed in the areas with positive answers which are sufficiently large

The computational time which is necessary for the complete processing of an image is reasonable. With a classical workstation (Sun4 SPARC) an image of size 256 × 256 is treated in 6 s (smoothing and normalisation of the image included). It is interesting that this algorithm could easily be installed on a more specialised machine as the major part of the operation is based on convolutions with kernels of size 5 × 5 or 8 × 8. Of course, the example of time given below has been obtained with an implementation using this property of the neural net. Using a net of six different machines, we are able to process one image per second and to present a 'live' demonstration.

In this paper, we assume that the faces are well oriented in the image. It is possible to eliminate this assumption by following an approach similar to that used for the scale problem. A net is trained to be insensitive to the precise orientation of the face. The network is scanned over several versions the image rotated by various angles (say every 20°).

This kind of segmentation algorithm can be applied to other problems where the object to be detected cannot be characterised easily by its outline or by classical primitives in image processing, such as car detection. Very little problem-specific modification is necessary: construction of a database of positive and negative examples suffices.

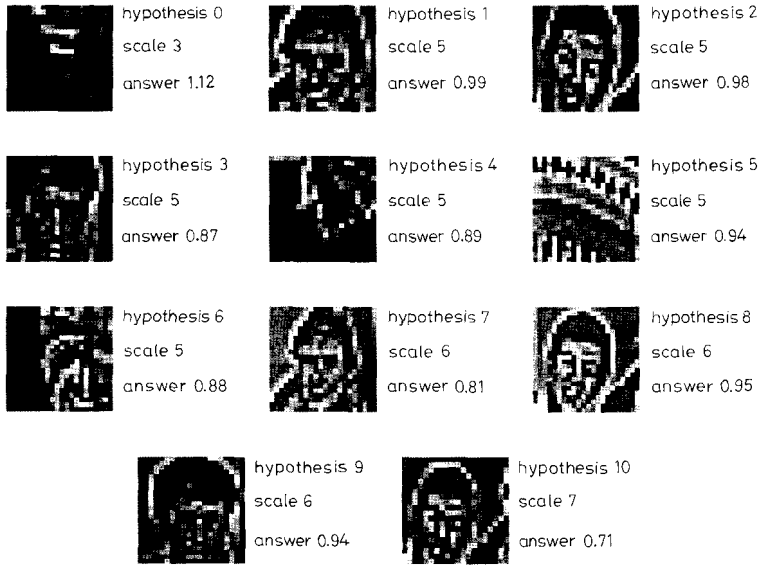

Fig. 7 *The hypotheses*

Fig. 8 *Localisation*
The hypotheses that have been retained at the end of the processing

6 References

1 CRAW, I., TOCK, D., and BENNETT, A.: 'Finding face features'. Presented at Second European conference on Computer Vision, April 1992
2 GOVINDARAJU, V., SRIHARI, S.N., and SHER, D.B.: 'A computational model for face location'. Presented at third international conference on Computer Vision, 1990
3 LE CUN, Y.: 'Modèles connexionnistes de l'apprentissage'. PhD thesis, Université Pierre et Marie Curie, Paris, France, 1987
4 LE CUN, Y., BOSER, B., DENKER, J.S., HENDERSON, D., HOWARD, R.E., HUBBARD, W., and JACKEL, L.D.: 'Backpropagation applied to handwritten zipcode recognition', *Neural Comput.*, 1990, **1**, (4)
5 PERRY, J.L., and CARNEY, J.M.: 'Human face recognition using a multilayer perceptron'. Presented at International conference on Neural Networks, January 1990
6 SAMAL, A., and IYENGAR, P.: 'Automatic recognition and analysis of human faces and facial expressions: a survey', *Pattern Recognit.*, 1992, **25**, (1), pp. 65–77
7 TURK, M.A., and PENTLAND, A.P.: 'Eigenfaces for recognition', *J. Cognitive Neuroscience*, 1991, **3**, (1), pp. 72–86
8 TURK, M.A.: 'Interactive-time vision: face recognition as a visual behaviour'. PhD thesis, MIT Artificial Intelligence Laboratory, September 1991
9 VINCENT, J.M., WAITE, J.B., and MYERS, D.J.: 'Precise location of facial features by hierarchical assembly of neural nets'. Proceedings of second international conference on Artificial Neural Networks, 1991, pp. 69–73
10 YUILLE, A.L., COHEN, D.S., and HALLINAN, P.W.: 'Feature extraction from faces using deformable templates', *in* 'Computer vision and pattern recognition' (1989)