In proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication, pp. 154-159. Washington D.C. March 1999

Face and feature finding for a face recognition system

Andrew W. Senior
aws@watson.ibm.com

IBM T.J.Watson Research Center
P.O.Box 704, Yorktown Heights, New York 10598-0218

Abstract

This paper deals with the problem of finding facial features in images, a problem which arises in face recognition and in a number of other applications, especially in human-computer interaction, which derive information from human faces. This paper describes a system for finding faces in images and for finding facial features given the estimated face location. The techniques, based on Fisher's linear discriminant and distance from feature space, are presented, and results are presented on faces from the FERET database. The paper further describes how feature collocation statistics can be used to verify feature locations and estimate the locations of missing features.

1 Introduction

The problem of face recognition has received increasing attention in recent years, with a growing number of applications being soluble by current technology. While error rates will probably never approach those of some other biometric systems, such as fingerprinting, face recognition is attractive because of its widespread use in non-automated systems, its user acceptability and the ease of acquisition.

Recently several face recognition systems have been produced both for verification of a user's identity — for such applications as computer logon, automated teller machines or cheque cashing — and also for identification in such applications as welfare fraud prevention, driving licence and voter registration.

Research in face recognition was classified by Brunelli & Poggio [1] into *feature-based* and *appearance-based* methods. However, the current literature seems to consist almost exclusively of appearance-based systems, which may be categorized as either *global* or *local*, according to whether they treat the face image as a single entity (such as the 'eigenface' approaches [2, 3]), or break it up into a number of local representations, as in the dynamic link matching method [4].

Though geometric features alone are not considered enough for face recognition, they are often used to supplement the discrimininating power of appearance features, cf. Dynamic Link Architecture [5] or constellations [6]. In addition, finding facial features is an essential part of most modern appearance-based methods, both local and global. Local methods use facial landmark features as anchor points with respect to which local appearance-based features are found [5]. For a global appearance-based method to be robust to facial expression changes, feature locations are also necessary. Generating a 'shape-free' representation requires identifying landmark features [7, 8]

Facial feature finding is also important for a number of other applications, not related to face recognition. The eye locations are important for eye-gaze detectors, and for iris recognition systems. Lip positions are important for lip reading to improve speech recognition in noise, and the positions of a variety of facial features can be used for determining affective state and interpreting the meaning of some speech acts, as well as determining head pose.

This paper describes a method of finding and tracking faces and facial features for use in any one of these applications, though with specific attention to its use as a feature locator for face recognition. The methods used are a combination of existing techniques including linear discriminants and distance from feature space.

2 Face finding

The first problem to be solved before attempting face recognition is to find the face in the image. In this work the face is found by a combination of methods, some of which are also used for feature finding. Face finding solves the important task of making face recognition translation, scale and rotation independent, and can provide good initial constraints on the location of facial features.

2.1 Colour-based segmentation

Many authors [9, 10] have reported methods of segmenting skin-tone regions from images, particularly for finding faces and hands. The colour-based segmentation used here is a simple scheme which requires no training, operates very quickly, and has been found to work well with different skin tones and under a variety of lighting conditions. Each pixel in the image is labelled according to whether it is close to skin tone or not. The scheme used is simple and fast, based on thresholds describing a cuboid in Hue, Chromaticity and Intensity space. Since the frame grabber used provides red, green and blue values, the HCI thresholds are transformed into limits on the RGB signal. This is done by determining the maximum and minimum values of the red signal for a given (green,blue) pair. The signals are quantized to 5 bits, so the result is two compact 1024-element tables.

The limits are calculated as follows, given intensity limits $I_{\min}, I_{\max}$ (intensity $= R+G+B$); hue limits $-\pi/2 < H_{\min} < 0, 0 < H_{\max} < \pi/2$; and chromaticity limits $C_{\min}, C_{\max}$. Given B, G and $M = \min(G, B)$, then:

$$I_{\min} - G - B < R < I_{\max} - G - B \quad (1)$$

$$G + B + C_{\min} < R < G + B + C_{\max} \quad (2)$$

$$R < \max(M, C_{\max} + 3M - G - B) \quad (3)$$

$$R > \min(M, C_{\min} + 3M - G - B) \quad (4)$$

$$\text{if } G > B,\ R > \frac{1}{2}\left(\frac{(G-B)\sqrt{3}}{\tan(H_{\min})} + G + B\right) \quad (5)$$

$$\text{otherwise } R < \frac{1}{2}\left(\frac{(G-B)\sqrt{3}}{\tan(H_{\max})} + G + B\right). \quad (6)$$

Given the pre-computed tables, for each pixel the red value is compared to the limits indexed by the green and blue values. If the value is between the limits, the pixel is labelled as being close to skin-tone. For efficient computation later, cumulative sums of the number of skin-tone pixels above and to the left of any given pixel are calculated and stored. This allows fast calculation of the number of skin-tone pixels in any rectangular region aligned with the axes using only three additions or subtractions.

In practice, when detecting faces, for each candidate location, the proportion of skin-tone pixels in the surrounding region is calculated, and the location is discarded if the proportion is below a predetermined threshold. For all the operations described below, the grey-scale image is used.

3 Fisher discriminant detection

A number of algorithms have been proposed for face detection based on the intensity pattern of the image.

Figure 1: A scene with its skin-tone classification.

The CMU system [11] uses a neural network. The neural network requires a significant number of calculations for each hypothesized face location. In designing the system described in this paper, a fast method requiring little calculation was desired.

The first pass of the face detector requires a simple dot product to determine if the location is 'face-like', by using a Fisher linear discriminant. The Fisher linear discriminant is trained by considering a database of face images, which have been labelled with eye and nose locations. A rectangular template size, $m \times n$, is chosen, and the template trained by extracting from each training image I an $m \times n$ sub-image, R containing a normalized face.

The new image, $R(I, m, n, x, \theta, E, k)$, is rectangular, with $m \times n$ pixels, and is centred on the nose location, x, with rotation, θ, to match that of the line joining the two eyes, and with scale k so that the inter-eye distances in all the re-sampled images are identical. If the inter-eye distance in an image is E, the width of the re-sampled image is chosen to be kE for some k (typically about 1.6). Furthermore, to remove pixels which may not be part of the actual face, the pixels at the corners of the re-sampled images are ignored, and the remaining pixels' intensities are considered as the elements of a vector. To remove the effect of overall lighting intensity, the mean of the vector's elements is subtracted from each element.

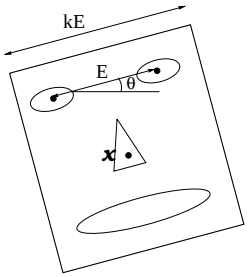

Figure 2: A diagram of a face showing the face candidate location parameters

Now the image of each face is represented as a zero-mean vector. The problem in hand is to classify quickly vectors generated in this manner from new images as being faces or non-faces. To this end, a Fisher Linear Discriminant [12, Ch. 5] has been used. To calculate the discriminant vector, a set of in-class vectors is found from the labelled face images in the database, and a set of out-of-class vectors is created by randomly generating locations, scales and rotations similar to those of true faces and resampling an image vector in the manner described above from images in the database.

Since the number of face samples may be smaller than the dimensionality of the face vectors that are generated, special care must be taken to prevent the problems of a singular within-class covariance matrix, as described by Belhumeur *et al.* [13]. To prevent this, the Fisher discriminant is calculated in the subspace spanned by the principal components of the joint face/non-face distribution. The dimensionality of this subspace is chosen to be less than the number of training images.

Having calculated the Fisher Discriminant Vector, any new candidate face hypothesis is tested for being 'face-like' by generating a sample image vector as described above, and finding the dot product. The resulting value is a measure of how 'face-like' the vector is, and can be thresholded to retain only those face-location hypotheses with a high score. While the two classes are by no means expected to be linearly discriminable, the discriminant gives a fast method for filtering out locations which do not resemble faces.

4 Distance from feature space

Pentland *et al.* [3, 14] have described the use of 'Distance From Face Space' (DFFS) as a method for face recognition. This is an aspect of work carried out on 'eigenfaces' [2, 3] where instead of finding a feature vector by projecting the image onto the first few eigenvectors $\mathbf{v}_i$ of the image covariance matrix, the distribution of the energy of the image over the eigenvectors is considered. If most of the deviation of the image away from the mean can be accounted for by the first few eigenvectors of the covariance matrix, then that variation is consistent with the variation seen in the faces in the training set, so the image is considered face-like. However, if much of the deviation is in the subsequent eigenvectors, then the variation is not consistent with the image being a face.

More formally, if the number of eigenvectors considered to model 'face-like' variation is f, then for some candidate image $\mathbf{x}$, $\mathrm{DFFS}^2 = |\mathbf{x}|^2 - \sum_{i=1}^{f} (\mathbf{x}.\mathbf{v}_i)^2$. This will be small for face-like images, and large otherwise.

This value can again be thresholded, but in practice, it has been found that a combination of this score with the Fisher linear discriminant gives a better identification rate. Currently the difference of the two is used, and in mug-shot images where a single face is known to be present, the location with the highest score is deemed the estimated face location.

5 Finding faces

In general, the scale, location and orientation of faces in an image are unknown. To be able to find faces in an image, a multi-resolution pyramid is used. The pyramid consists of successive sub-samplings of the original image at progressively lower resolutions. In each of these, every $m \times n$ rectangle of pixels is considered as a candidate face location, C. The ratio of successive image sizes is typically $\sqrt[3]{2}$ with the final level being the smallest resampling larger than $m \times n$.

If a colour image is available, the cumulative skin tone counts are tested first. The proportion of skin-tone pixels in the original image corresponding to the region of C is calculated and compared to the threshold.

Then the image vector corresponding to C is calculated, by removing the corner pixels and subtracting the mean. The dot-product with the Fisher linear discriminant is found, and if sufficiently high, the DFFS is calculated, and subtracted from the discriminant score to give a score for the location.

In this way all upright faces can be found, but if faces at any rotation are to be discovered, then the templates would have to be tried at many angles. However, for nearly all applications, it may be assumed that the face is approximately vertical. Furthermore the discriminant and DFFS both respond to faces at small rotations. Thus, the exhaustive search is only carried out for vertical faces, but when any face is detected, face candidates (generated by resampling the original image, not the pyramid) at small rotations from the vertical are also tested and accepted if they give a higher score than the vertical candidate. In a similar manner, the scale and location can be refined by resampling the original image at scales and translations close to the proposed candidate, but not included in the original pyramid.

6 Searching for features

Having found the face, and in a similar manner, the Fisher discriminant and Distance From Feature Space (also called DFFS and analogous to Distance From Face Space) can be used to locate features in the face. The features used can be varied, and may change depending on the application (*e.g.* face recognition, gaze

tracking, lip tracking, expression recognition) as well as the conditions (*e.g.* lighting direction, population to be recognized). The features used in this paper are shown in figure 3.

Figure 3: A diagram of a face showing, in white, the features to be located.

The scale and orientation are already known, since they are parameters used to generate the face candidate. For each face, the original image is re-sampled in the region of the face, at a scale proportional to the size of face found, and at the rotation which gave the maximum score. Thus, the search for features takes place in an image with predetermined scale with one vertical face at a known location.

Feature templates are generated by resampling the training images at the predetermined scale (given the eye-separation), taking rectangles of pixels centred on the features in question. In the test image, a simple strategy would be to test all possible locations, but anthropometric domain knowledge can be used to restrict the search, in the sense that the feature locations can roughly be predicted. From the training set, statistics are gathered for the normalized displacement of each feature, relative to the face centre, considered to be the nose location. The statistics collected are mean and variance in x and y directions, assuming that these are independent. The search is restricted to an ellipse s standard deviations wide about the expected position. Within each such search ellipse, the point with the highest score is retained. A low score may be considered as a failure, and the feature marked as not found.

7 Feature collocation

Although the feature location statistics provide a good guide as to the expected location of each feature, to allow for the full variation in face shapes, the search area is still quite wide, and it is possible to generate false matches in incorrect locations — for instance finding the end of an eyebrow instead of an eye corner. However, if a number of features is sought, many of these mistakes will become immediately apparent when the juxtaposition of the feature location estimates is considered. For instance, while the eye corners' displacements with respect to the nose can vary considerably across the population, in any given image, they will always lie close to a horizontal line. Similarly, the eyebrows must lie above the eyes and so on. Such collocation information gives us a valuable way of verifying the facial feature candidates. This section describes a method for determining which features in such constellations of features are plausible faces.

On the training set, the normalized displacement between each pair of features is recorded, and the mean and variance for each feature are calculated. Then, approximating the distribution of these displacements by diagonal covariance Gaussians, the log likelihood L of any given feature distribution can be found. If S is a set of features whose locations have been estimated:

$$L(S) = \sum_{i,j \in S} \left(-\log(\sigma_{ij}) + \frac{(x_i - x_j - \mu_{ij})^2}{\sigma_{ij}^2} + K \right) \quad (7)$$

While it is difficult to interpret the log likelihood itself, it is possible to use this to indicate which features are misplaced. Calculating the log likelihood for a set of features, and then successively calculating the log likelihood for all subsets of features leaving exactly one candidate out, the contribution to the likelihood of each feature can be seen.

$$\delta L_i = \frac{L(S \setminus i) - L(S)}{||S \setminus i||} \quad (8)$$

Any feature, i, whose removal causes a large increase δL_i in the log likelihood (normalized by the number of features contributing to the likelihood change) must be unusually situated in relation to the other features, and is discarded as a likely misdetection.

Having rejected one feature in this manner, the likelihood can be re-evaluated and features are iteratively discarded until the likelihood is high, indicating a consistent constellation of features.

7.1 Feature location prediction

The feature collocation statistics also give valuable information about the expected location of a feature if it is not found by the template matchers or if it is rejected based upon the collocation information. The location of a feature is predicted as the maximum

likelihood position, given the locations of the features which were located. Again it is assumed that the x and y components of the displacements are independently Gaussian distributed, and feature displacements are independent of one another. The maximum likelihood feature location $(\hat{x}_i, \hat{y}_i)$ is given by:

$$\hat{x}_i = \frac{\sum_j \frac{x_j + \mu_{ij}}{\sigma_{ij}^2}}{\sum_j \frac{1}{\sigma_{ij}^2}} \quad (9)$$

and similarly for $\hat{y}_i$ where j runs over all the features whose location (x_i, y_i) is known.

8 Face parameter re-estimation

The feature collocation statistics can also be used to obtain a better estimate of the face scale, location and rotation. If enough features are present, the locations can be used to estimate a maximum likelihood similarity transform to make them best fit the collocation statistics. This gives a better estimate of head scale, location and rotation with which to resample the face image before extracting a representation of the features for use in recognition.

All three applications of the feature collocation statistics have the advantage over some other methods that they work even if some or many of the features have not been detected (through failure of the detector, or through occlusion), or if some of the feature locations are incorrect.

9 Experiments

Experiments were carried out on a subset of the FERET [15] development set. 128 images from the `fa` set were used for training, and the corresponding 128 `fb` images used for testing. The training and test faces are marked up with a set of 19 facial features, as shown in figure 3. They are: pupil centres (2), eye corners (4), nose, nostrils (2), nose corners (2) and bridge, eyebrow endpoints (4), mouth corners (2) and top lip centre,

The first experiment measures the face detection accuracy. Since the FERET dataset is in black and white, the colour face segmentation scheme was not used. The true face location and pose are determined from the ground truth nose and eye locations. Face detection is considered successful if the centre location is within $0.25E$ of the ground truth nose position, E being the ground-truth eye separation, roughly 45 pixels for this dataset.

The second experiment measures the accuracy of the feature detection, given the correct face location. A feature is considered to be correctly located if found within $0.1E$ of the correct location. Table 2 shows the aggregates for all features and table 3 shows the detection rates broken down by feature. The area of the image searched for a feature is an ellipse of radius two standard deviations. This amounts to an average of 131 pixels per feature. Using the linear discriminant to filter candidate locations means that only 27 pixels per feature are searched with DFFS. Table 2 summarizes the results across all the test set, and shows the benefit of using the collocation statistics for removing mis-detected features, and for predicting the correct location of those features.

Error rate %	Distance	Rotation	Scale
2.4	0.08	2.0	1.09

Table 1: Face detection errors. The three error measures are evaluated on those faces deemed to be correctly detected. Error distance is quoted in multiples of the eye-separation, E, rotation error is the mean absolute rotation error in degrees, and scale error is the geometric mean of the ratio in scales between the estimate and correct value.

Average number of features correct		
Before pruning	After pruning	After inference
14.84	14.35	15.43

Table 2: Feature detection rates, from 19 features.

Finally, an experiment was conducted in which the face pose was estimated using the face finder and the feature locations estimated with this pose. Finally the feature collocation statistics were used to reestimate the pose. The scale error was reduced by 2.3% and the location error reduced by 39%.

The results highlight a number of aspects of the system. First, the pruning method is effective in reducing the number of erroneous feature detections, pruning out an average of 1.74 features per face, but reducing the number of correct features per face by 0.49. This trade-off is worth while, since inferring the missing features using the collocation statistics, the total number of correct features is increased. The eyebrow features are found to be hard to detect, because of great variability in shade and shape. The collocation statistics prove useful in re-estimating the face pose.

10 Future work

In future work, there are a number of methods to refine the system. The current methods of finding the pose and rejecting outlier features will be compared with other methods such as RANSAC [16]. It will also be seen if the search can be improved by enforcing symmetry constraints, and applying the collocation

Feature	Mis-detection rates (%)		
	Initial	Pruning	Inference
R.O.Eyebrow	29	15	23
R.I.Eyebrow	49	44	44
L.I.Eyebrow	23	20	24
L.O.Eyebrow	46	31	34
R.O.Eye	21	15	17
R.Eye	8	4	7
R.I.Eye	20	9	9
L.I.Eye	15	6	8
L.Eye	13	6	7
L.O.Eye	15	10	13
Nose Bridge	28	19	23
R.Nose	7	6	6
R.Nostril	5	3	3
Nose	3	3	3
L.Nostril	0	0	0
L.Nose	6	4	5
R.Mouth	13	11	15
L.Mouth	19	15	19
Upper Lip	18	12	14

Table 3: Feature detection error rates, for the 19 features used, using the Discriminant/DFFS feature detector, after pruning based on feature collocation and after re-estimating pruned features.

constraints more strongly during search. A coarse-to-fine feature-finding method is currently being tested.

Acknowledgments

Thanks are due to Jon Connell for the original HCI skin-tone segmentation extended in this paper.

References

[1] Roberto Brunelli and Tomaso Poggio, "Face recognition: Features versus templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042–1052, October 1993.

[2] M. Kirby and L. Sirovich, "Application of the Karhunen-Loève procedure for the characterization of human faces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103–108, 1990.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuro Science*, vol. 3, pp. 71–86, 1991.

[4] Laurenz Wiskott and Christoph von der Malsburg, "Recognizing faces by dynamic link matching", *in Proceedings of the International Conference on Artificial Neural Networks*, pp. 347–352, 1995.

[5] Laurenz Wiskott, Jean-Marc Fellous, and Norbert Krüger, "Face recognition by elastic bunch graph matching", Technical Report IR-INI 96–08, Buhr-Universität Bochum, Institut für Neuroinformatik, April 1996.

[6] Karin Sobottka and Ioannis Pitas, "A fully automatic apporach to facial feature detection and tracking", in Josef Bigün, Gérard Chollet, and Gunilla Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, vol. 1206 of *Lecture Notes in Computer Science*, pp. 77–84. Springer, March 1997.

[7] Ian Craw and Peter Cameron, "Face recognition by computer", in David Hogg and Roger Boyle, editors, *Proceedings of the British Machine Vision Conference*, pp. 498–507. Springer Verlag, September 1992.

[8] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic interpretation and coding of face images using flexible models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 743–756, July 1997.

[9] Jean-Christophe Terrillon, Martin David, and Shigeru Akamatsu, "Automatic detection of human faces in natrual scene images by use of a skin color model and of invariant moments", *in International Conference on Face and Gesture Recognition*, number 3, pp. 112–117. IEEE, April 1998.

[10] T. Darrell, G. Gordon, J. Woodfill, and M. Harville, "A virtual mirror interface using real-time robust face tracking", *in International Conference on Face and Gesture Recognition*, number 3, pp. 616–621. IEEE, April 1998.

[11] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Human face detection in visual scenes", Technical Report CMU–CS–95–158R, School of Computer Science, Carnegie Mellon University, November 1995.

[12] Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, 1973.

[13] P.H. Belhumeur, J.P. Hespanha, and D. J. Kriegmand, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, July 1997.

[14] B. Moghaddam, W. Wahid, and A. Pentland, "Beyond eigenfaces: Probabilistic matching for face recognition", *in International Conference on Face and Gesture Recognition*, number 3, pp. 30–35. IEEE, April 1998.

[15] P. Jonathon Phillips, Hyeonjoon Moon, Patrick Rauss, and Syed A. Rizvi, "The FERET September 1996 database and evaluation procedure", in Josef Bigün, Gérard Chollet, and Gunilla Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, vol. 1206 of *Lecture Notes in Computer Science*, pp. 395–402. Springer, March 1997.

[16] A.H. Gee and R. Cipolla, "Determining the gaze of faces in images", Technical Report CUED/F-INFENG/TR 174, University of Cambridge, Department of Engineering, Trumpington Street, Cambridge CB2 1PZ, England, Mar. 1994.