

Recognizing faces in broadcast video

A.W.Senior

IBM T.J.Watson Research Center,
Yorktown Heights, NY 10598, USA.

aws@watson.ibm.com

Abstract

Face recognition has recently attracted increasing attention and is beginning to be applied in a variety of domains, predominantly for security, but also for video indexing. This paper describes the application of a face recognition system to video indexing, with the joint purpose of labelling faces in the video, and identifying speakers. The face recognition system can be used to supplement acoustic speaker identification, when the speaker's face is shown, to allow indexing of the speakers, as well as the selection of the correct speaker-dependent model for speech transcription. This paper describes the feature detection and recognition methods used by the system, and describes a new method of aggregating multiple Gabor jet representations for a whole sequence. Several approaches to using such aggregate representation for recognition of faces in image sequences are compared. Results are presented showing a significant improvement in recognition rates when the whole sequence is used instead of a single image of the face.

1 Introduction

Face recognition is an intriguing and challenging problem for a number of reasons. Many researchers are interested in discovering clues as to how people can recognize faces, while others are encouraged by the wealth of applications for an automatic face recognition system. Thus far, most systems have treated face recognition as one more biometric, suitable for security applications, for physical access control and computer logon, or in database lookup in databases of face images, such as those used by the police and passport or driving licence authorities.

This paper, however, concentrates on an application that is uniquely suited to face recognition — that of recognizing the people depicted in broadcast video. There are a number of applications of a system that can accomplish this task, the main one being that of video indexing [6]. In this scenario, as video is ingested into a database, for instance from live news feeds, a database of 'interesting' faces is com-

pared with all the faces found in the video, and the identities (as well as number) of the faces matching the database are recorded as indices for future searches. This system could be applied with a large population database, or on specific domains with small databases, such as world leaders, reporters and anchors for news programs; members of the participating teams for a sports match; or cast members of a given situation comedy. An extension of such a system would automatically cluster unknown face appearances by identity for similarity searching or easy hand labelling.

A second application is to identify speakers in a given video clip, so that appropriate speaker-dependent speech recognition models can be used for high accuracy speech decoding. The face recognition system described in this paper has been combined with an acoustic speaker identification system [4], to give a fast audio-visual speaker identification that is robust to acoustic noise conditions. This application of face recognition does have limitations not present in the former application, in that video often contains faces not belonging to the speaker — when there are multiple faces, or when there is a voice-over. In these situations, the face recognition system must either be limited to refining answers given by acoustic methods, or other methods verifying that the video face is the current speaker must be applied before using the results of the face recognition algorithm.

This paper describes a complete system for face recognition, applicable to both these tasks, that has been fully automatically trained and tested on a database of real broadcast television video sequences.

2 Face recognition

This section describes the methods used for face detection, feature location and face recognition used by the recognition system. Earlier versions of the face and feature location system, working on still images have been described elsewhere [7].

3 Face detection

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. Face finding in colour video raises a number of special issues that need to be handled differently to the face finding in still black and white mug-shot images previously recognized by this system.

Firstly, since the signal is in colour, skin-tone segmentation can be used to narrow the search for faces to only those regions which contain a high proportion of skin-tone pixels. Secondly, the number of faces is unknown. A mug-shot image is guaranteed to contain a single face, but a frame of broadcast video may contain any number of faces, including zero. Further, the faces in mug-shot images are generally highly constrained in scale, position and orientation. Finally, since video is made up of what might be termed ‘piecewise continuous’ shots, information about the face locations in a given frame generally conveys a large amount of information about the faces present in the successive frames. The system described here can handle, or exploit, all of these differences.

3.1 Face detection in an image pyramid

To find all the faces in an image can be viewed as a classification problem. In general, at any position in the image a face could exist at any scale or rotation. The problem is simplified by looking for only the central square of a face, defined as a square centred on the nose, angled so the top is parallel to the line joining the eye centres, and with sides some fixed factor k longer than the separation between the eyes. Any square region of the image is termed a ‘face candidate’ and the classification algorithm must determine for all of these, whether it represents a face or not. The problem can be simplified significantly by specifying limits, based on domain knowledge, on the scales of faces that can be found in the video, or are determined to be of interest, as well as limiting the search to faces close to the vertical.

A template size ($m \times m$ e.g. 11×11) is chosen, and the image is sub-sampled such that the smallest face to be detected would be the same size as the template. This image is re-sampled at progressively lower resolutions until the largest possible face would be the same size as the template in the final image. This sequence of sub-sampled images is termed an image pyramid. Any face in the original image which falls into the range of scales of interest, should correspond to a square region in one of the pyra-

mid’s images, whose size is the same as that of the template (to within some scale tolerance which determines the sub-sampling ratio). Now the problem of finding faces in the image is that of determining whether $m \times m$ squares of pixels in the image pyramid are faces or not.

This two-class decision process is carried out by a combination classifier, which executes hierarchically to quickly filter out regions unlike faces, and spends more time on regions which are less clear. The first stage of the process is the colour segmentation, which simply determines if the proportion of skin-tone pixels [7] is greater than some threshold. Subsequently candidate regions are given scores based upon Fisher linear discriminant and Distance From Face Space, which are combined into a joint score. All candidate regions exceeding a threshold are considered to be faces, after applying constraints such as no two faces may overlap.

The parameters of face candidates that are determined to be faces are subsequently refined by searching at nearby scales, locations and rotations not considered in the initial search, and among these variations, picking that with the highest score.

3.2 Face tracking

Video face tracking is currently carried out using a simple search technique, although more sophisticated algorithms are available [11]. The initial frames of video are searched exhaustively for faces, using the image pyramid described above, until a frame with faces in it is found. Then those faces are tracked, and an exhaustive search is conducted every few frames, or if the faces are lost by the tracking algorithm. Face tracking uses face position differences from successive frames to calculate a velocity vector and thus to predict the location of the face in the next frame. The face is searched for in a small region close to the predicted location, and at similar rotations and scales.

4 Searching for features

A previous paper [7] has already described how the Fisher discriminant and distance from feature space (DFFS) can be used along with prior feature location statistics to determine the location of facial features. This paper describes an enhancement to this method that enables features to be found more accurately and faster than possible with the previous method. The crucial difference of the new method is to search for features hierarchically. Instead of searching for the facial features directly in the face image, a few ‘high-level’ features (eyes, nose, mouth) are located, and then the 26 ‘low-level’ features (parts of the eyes, nose, mouth, eyebrows etc.) are located relative to the high-level feature locations. The feature locations at both levels are determined

using the same combination of prior statistics, linear discriminant and DFFS.

The first stage in the search is to normalize the face image. Given the location, scale and rotation parameters of the detected face candidate, a normalized sub-image is re-sampled from the original frame, so that the eyes are in a horizontal line with a fixed separation, E . The approximate locations of the high-level features are known from statistics of mean and variance (relative to the nose position) gathered on a training database. The discriminant/DFFS templates are used to score each location with high prior probability of containing a given feature. Typically an area representing around 2 standard deviations is searched. Within the search region, the location with the highest score is deemed to be the location of the feature.



Figure 1. A diagram of a face showing, in white, the automatically located features.

The locations of the low-level features relative to the nearest high-level features are also measured on the training set, and the displacement statistics recorded. Given the high-level feature location estimate, a search area for each low-level feature is determined (again corresponding to typically 2 standard deviations) and searched using a template. This search is carried out on an image of the face re-sampled in the same manner as the high-level feature search, but at higher resolution (*i.e.* mapping the eyes to be further apart).

As before [7], the feature locations determined by this method are subject to verification using collocation statistics. This is implemented as a pruning of features whose location is inconsistent with the other features detected, as determined by a probabilistic score. Further, the collocation statistics can be used to infer the correct locations of these features. The results quoted later indicate the benefit of this pruning and inference combination.

In addition to the 26 features located visually with trained templates, a number of feature locations are deter-

mined for features which are not visually well-defined, yet are useful for determining identity. In particular, cheeks and forehead tend to be homogenous areas, but could vary dramatically according to the presence of facial hair and the height of the hairline. Training local templates to locate these features would be fruitless, and they are instead defined geometrically with respect to other features. The cheek locations are defined to be the midpoints of the mouth corner and outermost eye corner for each side of the face, and the forehead is the point defined by $2B - N$ where B is the location of the nose bridge and N is the location of the nose tip. These features are left undefined if either of the features required for their calculation were not located. The addition of these geometric features gives a total of 29 feature locations that can be found in a face.

The failure of feature detection is good indicator of the failure of face detection. If the features can not be found reliably in a face candidate — when the sum of the feature-detection scores for the most salient features falls below a threshold — then it is rejected as being a false alarm of the face detector.

4.1 Feature detection experiments

The first set of experiments illustrates the improvement in feature location accuracy achieved by using the hierarchical feature location method. This test was carried out on a subset of the FERET [5] development set. 128 images from the f_a set were used for training, and the corresponding 128 f_b images used for testing. The training and test faces are marked up with a set of 19 facial features, as shown in figure 1. 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 other 10 features are less stable, and are not used in this experiment but assist in identification (hairline, chin, cheeks (2), forehead, ears (2), lip points (3)).

This experiment assumes the correct face location, as given by hand labelling. A feature is considered to be correctly located if found within $0.1E$ of the correct location. Table 1 shows the aggregates for all features and table 2 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 86 pixels per feature. Using the linear discriminant to filter candidate locations means that only 35 pixels per feature are searched with DFFS. Table 1 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.

Average number of features correct		
Before pruning	After pruning	After inference
15.3	14.8	15.6

Table 1. Feature detection rates, from the 19 most reliable features.

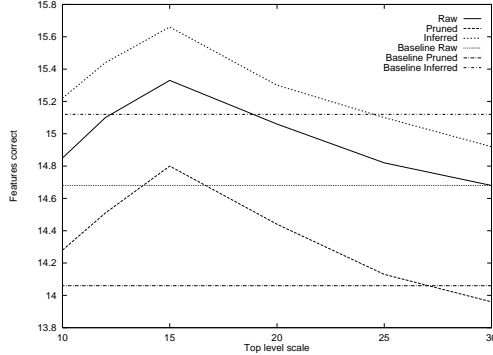


Figure 2. Feature detection rates varying the hierarchy parameters. The baseline (horizontal) results are shown for a system using no hierarchy. Separate curves are shown for the raw, initial results (middle), results after pruning based on collocation (lowest) and after inference (highest).

5 Face recognition

Face recognition requires storage of an identity template — the data representing the face in question. For this work, a template of local identity information has been used, in contrast to global identity templates, used for instance in Eigenface systems [8]. A local template has been chosen because of its greater robustness to facial image changes caused by effects such as lighting, expression, or facial appearance change (glasses, beard, haircut etc.). In this case a simple Gabor jet model has been used, similar to that used by Wiskott and von der Malsburg [10].

For this representation, a feature vector is generated for each of the 29 facial features located above. The feature vectors consist of 40 complex elements each, representing the filter responses of Gabor filters with 5 different scales and 8 different orientations, centred at the estimated feature location.

Templates from two different faces are compared using

Feature	Mis-detection rates (%)		
	Initial	Pruning	Inference
R.O.Eyebrow	19	13	22
R.I.Eyebrow	40	32	38
L.I.Eyebrow	13	5	8
L.O.Eyebrow	32	27	32
R.O.Eye	13	5	8
R.Eye	9	2	4
R.I.Eye	13	3	7
L.I.Eye	9	2	5
L.Eye	4	1	3
L.O.Eye	6	3	8
Nose Bridge	27	16	17
R.Nose	8	2	3
R.Nostril	6	2	3
Nose	2	2	2
L.Nostril	2	1	2
L.Nose	1	0	2
R.Mouth	13	9	15
L.Mouth	17	11	16
Upper Lip	19	12	15

Table 2. 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.

a similarity metric

$$S(\mathbf{a}, \mathbf{a}') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j (a'_j)^2}} \quad (1)$$

which compares each feature \mathbf{a} in one face with the corresponding feature \mathbf{a}' in the other face. If either feature is not located, the feature vector and thus its similarity, $S(\mathbf{a}, \mathbf{a}')$, to the corresponding feature vector in the other face are undefined. Similarity scores for the different features are simply averaged across all those features where the similarity was defined.

Robustness to image changes caused by appearance change etc. can be achieved by comparing only subsets of the features in question. Two faces whose eye features matched well, but whose mouth regions did not could be interpreted as being the same person with a different mouth expression, occlusion of the mouth or with changed facial expression. A global representation of identity would have difficulty recognizing this local similarity.

5.1 Face recognition experiments

The second experiment uses the entire system to recognize faces in broadcast news video. The training data consists of 76 clips of video from CNN & CSPN broadcast news footage, digitized in MPEG2 format. The clips are selected by hand with the criterion of containing a single speaker in the acoustic track. In practice this means that the majority of each clip has a single talking head centred in the screen, but some clips show more than one face and many begin or end with images from a different visual shot. Such sequences might be automatically generated with a speaker change detection algorithm [1].

For the initial experiments described here, face recognition was carried out on a face image from a single frame of the video. This *key face* was chosen to be the face detection instance for which a heuristic ‘face and feature location score’ was maximized. This score is a weighted sum of the face and feature detection scores with the collocation score. It thus indicates when a ‘good’ face match was obtained, with ‘good’ feature matches within it, in a distribution close to those observed in the training set.

The Gabor jet coefficients for each such key face were generated and stored in a training database. Similar key faces were generated on 155 different video clips showing the same people found in the training set. For each of the test faces, the database face with the maximum similarity was found, and deemed to be the recognized face.

Method	Correct (%)
Key face	70.6
Sequence, batch (Bhattacharyya)	81.0
Sequence, batch (diagonal d')	87.1
Sequence, frame (diagonal likelihood)	75.6

Table 3. Face recognition rates on video sequences, using different recognition methods, based on key-frame or whole-sequence models.

The recognition method described above risks choosing a frame, either in training or testing, that is not representative of the person, and thus giving a poor match. In practice it is found that as many as 5% of the key faces for the sequences given are faces not matching the sequence label — faces of people other than the speaker who happen to appear in the sequence. Other key-faces are entirely erroneous, being areas of background with colour and texture similar to the face.

If the training stage can be supervised allowing the manual selection of appropriate frames, of course the recognition accuracy can be improved. However, the unsupervised

system can be improved by using the redundancy in video. In a video sequence, a large number of frames is available, conveying more information about the person, possibly with a variety of scales, head poses, facial expressions and lighting conditions. Making an aggregate model of all the faces in the training sequence will give a better representation of the person’s facial appearance, and will reduce the risk of choosing a bad frame. Similarly, in testing, all the frames of the test sequence should be compared against the model, to find the model which matches the whole sequence best. Edwards *et al.* [2] have shown significant improvements by integration of evidence from video sequences.

Initial experiments have been carried out using such a whole sequence model, implemented as follows: The Gabor jet coefficients are generated as before, but in this case they are generated for every frame of the video sequence in which a face is found. The mean and diagonal covariance of the coefficient magnitudes over all the frames of the sequence are calculated and stored. Where multiple faces or false-alarms are discovered, the representations are aggregated into the jet statistics, on the assumption that tracking errors and spurious faces will form a small number of the face detections. In sequences where several faces are present throughout, face detection will only be able to narrow the choice to those few, but using face detection alone the identification is necessarily ambiguous.

Sequences are compared in batch mode by finding a distance between training and test distributions. Experiments have been carried out with the Bhattacharyya distance [3, p188] and the simpler d' distance. The latter has been implemented using only diagonal covariances and, in addition to requiring much less computation, has been found to perform better on this task.

Alternatively, recognition can be carried out frame-by-frame using a training set constructed from the jet coefficient statistics. In this case, for each face found in a sequence, its likelihood given each of the training set models is calculated, assuming the coefficients are Gaussian-distributed. For a sequence, the likelihoods are summed, and compared at the end of the sequence, taking the maximum likelihood training model as the correct answer. For speed of computation, diagonal covariance matrices are used.

5.2 Real-time implementation

The Experiments described above have been conducted on MPEG2-encoded data, tracking the faces from frame to frame. Conducting full face tracking, feature detection and face representation on every frame of a video encoded at 30 frames per second runs slower than real time as currently implemented. However, even without efforts at speed-up, the system can be run in real-time, by tracking at a lower

frame rate. Since in the limit, the face only needs to be detected and encoded for a single frame, the frame rate could be very low, though in some applications high latency could be a problem. The current live system tracks faces at 7 frames per second on a 400MHz Pentium, in only 22% of the CPU power, operating on QCIF images. With recognition on every frame, this falls to two frames per second. In practice, even to reap the benefits of multi-frame information integration, this is an acceptable frequency for recognition. The code operating on the MPEG2 video, could be speeded up significantly by only carrying out feature detection and recognition every few frames, instead of every frame as at present.

6 Conclusions and future work

This paper has described a face recognition system and its application to speaker recognition in broadcast news video. Results show practical results with a medium-sized training database of faces. Using data from a whole sequence rather than a single face instance makes the system much more robust to tracking and detection errors, and also to sequences in which more than one face is present.

The approach described above shows encouraging initial results for a wholly automatic system operating on real world data (actual broadcast video). Such results are practical for certain simple video indexing problems, such as labelling programs with a small cast. However, a number of simplifications have been made in the initial implementation of the system, and further research is needed to optimize and improve upon these methods.

The heuristic for determining the 'best' frame is also very simple and could be improved upon, since the key frames still represent faces with closed eyes, non-full-frontal faces etc. This could be combined with the full-sequence representations, by weighting the frames within a sequence according to the confidence in the face and feature detection. Work is also continuing to increase the size of the database.

Further speed improvements could be made by operating in the compressed domain [9] of the MPEG video, or improving the frame-grabber code to make use of all the CPU time.

Acknowledgments

Thanks are due to members of the IBM speech group for their assistance in providing the data on which these experiments were carried out.

References

- [1] H. S. M. Beigi and S. H. Maes. Speaker, channel and environment change detection. In *World Congress on Automation*, April 1998.
- [2] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Improving identification performance by integrating evidence from sequences. In *Proceedings of Computer Vision and Pattern Recognition*, pages 486–487, 1999.
- [3] K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.
- [4] C. Neti and A. W. Senior. Audio-visual speaker recognition for broadcast news. In *DARPA Hub 4 Workshop*, March 1999.
- [5] P. J. Phillips, H. Moon, P. Rauss, and S. A. Rizvi. The FERET September 1996 database and evaluation procedure. In J. Bigün, G. Chollet, and G. Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, volume 1206 of *Lecture Notes in Computer Science*, pages 395–402. Springer, March 1997.
- [6] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in video by the integration of image and natural language processing. In *Proceedings of IJCAI*, pages 1488–93, 1997.
- [7] A. W. Senior. Face and feature finding for a face recognition system. In *Second International Conference on Audio- and Video-based Biometric Person Authentication*, March 1999.
- [8] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.
- [9] H. Wang and S.-F. Chang. A highly efficient system for automatic face region detection in mpeg video. *Transactions on Circuits and Systems for Video Technology*, 7(4):615–628, August 1997.
- [10] L. Wiskott and C. von der Malsburg. Recognizing faces by dynamic link matching. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 347–352, 1995.
- [11] J. Yang and A. Waibel. A real-time face tracker. In *WACV'96*, pages 142–147, 1996.