

 Open access • Book Chapter • DOI:10.1007/978-1-4471-3201-1_52

Face Recognition by Computer — Source link

Ian Crow, Peter Cameron

Institutions: University of Aberdeen

Published on: 01 Jan 1992 - British Machine Vision Conference

Topics: Facial recognition system

Related papers:

- [Eigenfaces for recognition](#)
- [Application of the Karhunen-Loeve procedure for the characterization of human faces](#)
- [Eigenfaces vs. Fisherfaces: recognition using class specific linear projection](#)
- [View-based and modular eigenspaces for face recognition](#)
- [Human and machine recognition of faces: a survey](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/face-recognition-by-computer-2jkwoicp5n>

Face Recognition by Computer

Ian Craw and Peter Cameron

Department of Mathematical Sciences *

University of Aberdeen AB9 2UB, Scotland

Abstract

We describe a coding scheme to index face images for subsequent retrieval, which seems effective, under some conditions, at coding the faces themselves, rather than particular face images, and uses typically 100 bytes. We report tests searching a pool of 100 faces, using as cue a different image of a face in the pool, taken 10 years later. In two of three tests with different faces, the target face *best* matches the corresponding cue.

Our codes are obtained by texture mapping the face image to a standard shape and then recording both shape and texture. Principal component analysis both reduces the data to be stored, and also improves its effectiveness in describing the face itself, rather than a particular image of the face.

1 Recognising Faces

We are all familiar with the problem of identifying a person cued only with a picture of their face. The difficulty that people have is an indication of the problems that arise when seeking a computer-based solution. The face has to be identified from many different angles, under different lighting conditions, and with different facial expressions. Greater difficulty is usually found if the face is seen in an unusual context, or is a different age from the familiar one.

It is thus not surprising that there have been few successful automatic face recognition devices, despite the commercial potential, particularly for security-based applications. Almost all attempts at computer face recognisers have explored one of two very different strategies to do this.

- One method proceeds via edges to feature recognition, and thus to a full “understanding” of the face as defined by some model, which is then used to match against a database. Each face to be recognised must have a distinct, usually hand-built model, which may just contain relative feature locations, or may contain more detail. An early success following this strategy was by Kanade [12].
- The other approach, via neural nets, goes directly to the desired identification. Specific instances of recognition are provided during the training

*PC is supported by an SERC studentship. Both authors acknowledge the help of other members of the Aberdeen “Faces” group, and particularly that of Andrew Aithison, whose software for placing points aided much of the work described here.

phase, and there is no attempt to identify any abstract intermediate representation held by the "hidden units". Such schemes have successfully recognised single faces [19], and distinguished between a pool of faces cued from a degraded copy of an image in the pool [14], [16]. Training can be a problem when working with a single copy of many different faces if the cue is a new image, but Starkey and Aleksander report some success here (cf the 1992 version of [18]).

Recently, hybrid methods have emerged, which use a preprocessing stage to register the face image and provide more homogeneous input to a recognition component. Examples of this approach include Kirby and Sirovich [13], representing faces using principal components; Cottrell and Fleming [5], attempting net based recognition; Turk and Pentland [22] with recognition based on principal components; and Brunelli and Poggio [4], performing gender classification.

In this paper, we extend this hybrid approach by making the preprocessing stage explicit and, in principle, automatic. We also recognise that more preprocessing, even to the stage when the input image no longer resembles the raw image, brings additional power. Other hybrid methods proposed recently, such as [20] and [9] have a less elaborate coding stage.

Our model of face recognition consists of a number of face images in a *pool*, together with a new image, the *cue*. The interesting case occurs when one or more images of a face (the *targets*) are in the pool, and a different image of that face is used as cue. The aim is to retrieve from the pool those images which best match the cue. This model includes the case of *single person* recognition, in which all the images in the pool are of a single face; however, in what follows, all our tests are done with a single image as target, in a pool of 100 faces. Of course this model says nothing about how the cue and each member of the pool are matched; the major novelty of our work lies in the choice of coding scheme to permit this matching.

2 Coding Faces

Our method of coding a face is a two stage process, reflecting the hybrid nature of the processing. The face is first located in the image, and features found well enough so that a mesh can be drawn as in Fig. 1, with each vertex or *control point* at a known position — the left corner of the mouth, or the centre of the eye etc. Once the control points are located, we distort the face by moving each control point to its position on the average face, and allow the texture to follow. This "full anticaricature" [2] is also shown in Fig 1.

For the pilot studies we describe here, the individual control points are located manually. Making this information available prior to processing means that a lot of knowledge about the shape of each face has been given. In a parallel project, software (FindFace) is being developed, capable of locating these points automatically [7]. At present FindFace finds the control points shown in Fig. 5; these points were needed for a different demonstration, and the same methods yield the control points required here. In a "typical" set of 64 images, the eyes were located accurately in every image, and most features can be found with a high degree of reliability. Even in much more difficult images the overall location is usually correct. Work continues to improve FindFace's

robustness and integrate it to provide completely automatic face coding. We regard this capability as an important part of the work described here.

Once the control points have been located, the image is texture mapped to a standard position, as shown in Fig. 1. We refer to the resulting images as *shape-free faces*, because corresponding features occur in corresponding positions in each image, and we have removed the shape information. A shape-free face is no longer a normal square image, but at each point within the standardised shape, we associate, by texture mapping, a grey level.

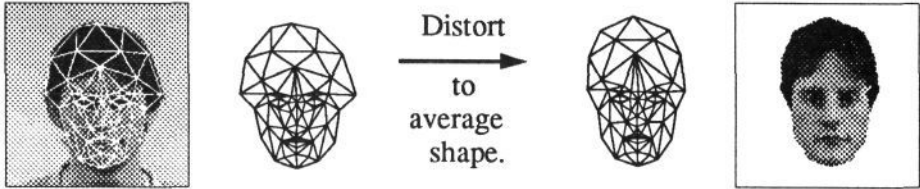


Figure 1: Control points are located and the mesh mapped to the average shape; the grey levels follow giving a shape-free face or texture vector.

We now describe the two codes we use subsequently for both matching and retrieval. The first of these, the *shape vector*, consists simply of the locations of each of the control points. The second, which we refer to as the *texture vector*, consists of the vector of grey levels used to texture the corresponding shape-free face. Apart from distortions introduced by imperfect texture mapping, the original face may be reconstructed from this representation. Our proposed coding scheme gives two distinct methods of recognition:-

Texture: since all the shape-free faces are “registered”, so that like features occur in the same place in each image, they form an appropriate set of images on which to perform standard template matching, using the cue as template. Although the template may have 30K pixels, this is practicable for small pools. We describe in Section 5 how this difficulty can be overcome; indeed we can not only drastically reduce the size of this code, but improve performance at the same time.

Shape: another way uses the shape vector itself (strictly, relative positions from which global position, scale and orientation effects have been removed), and searches the pool for those faces which best match. Similar faces have face features in the same relative positions; and we have a simple anthropometric system.

For convenience we have described the texture vector as that arising from mapping to the shape of the average face. This is familiar from the work on caricaturing of Benson and Perrett [2], who distort individual faces *away* from the average by an amount proportional to the deviation of the particular face from the average. In fact there is no need in our methodology to use an average; we simply need each face to be distorted to the *same* shape. We are experimenting with old idea of Baron [1], in which proportionately larger areas are devoted to those parts of the face, such as the eyes, which are most important for recognition. A variant is to remove the hair completely from

consideration; although hair is important for recognition, it varies a great deal over long periods of time. We present results with such a template in both Table 1 and Table 3.

3 Template Matching

In this section we describe recognition results obtained using just the texture vector as a code on which to perform matching. A *pool* of 100 images was used, all drawn from the Aberdeen face database. This collection of face photographs was made in 1982, in order to test a methodology for mugshot retrieval, and the collection was a simulation of such a mugshot database [17]. Subjects were photographed in a uniform way, ensuring that differences such as background and lighting were effectively eliminated. The images were subsequently digitised in as uniform a manner as possible with location and scale fixed so that each left eye and each right eye occupied a common position. The images were subsequently normalised to the same mean intensity, control points were located on each image by hand, and the corresponding texture vector was stored.



Figure 2: *Modern images of Ian, Ken and Harry, used as cue to search the database.*

In order to test recognition, modern (1991) images were obtained of three of the faces in the original collection. These images (we refer to them as Ian, Ken and Harry) were collected under very different conditions, in a different laboratory, with the lighting in use there. No attempt was made to imitate the conditions that obtained in the earlier collection. In Fig. 2 we show these images, and in Fig. 3, the originals, together with the images of three other faces from the pool to give an idea of the variation between images.



Figure 3: *Six images from the pool of 100, including images of Ian, Ken and Harry taken nearly ten years before those in Fig. 2.*

The cue images were distorted to shape-free form, and the intensity normalised as in the pool. A matching test was then performed using each of the

modern images in turn as a cue. The dot product with each image in the pool was calculated, and the magnitude used as (naive) notion of similarity to rank the images in the pool as matches for the cue image. We reproduce in Table 1 the rankings of each of the target images, as matches for the corresponding cue. We report also the rankings obtained in a similar test, in which the portion of the shape-free face used as a template was restricted by excluding the hair.

It is hard to present results in a way which guarantees that bias has been excluded, but the distractors in the pool are believed to be “random”. All the tests we report here have been with a fixed pool consisting of faces photographed under very similar conditions in order to avoid difficulties in knowing whether a match occurred for accidental reasons associated with the image, or because the face itself was recognised. At the very least, our results suggest the potential of a shape-free face as a code for recognition. We return to this in Section 5 after presenting results matching on shape alone.

Cue face	Ian-now	Ken-now	Harry-now
Complete Face	1	39	18
Ignoring Hair	1	13	22

Table 1: *Template matching on a pool of size 100. The table gives the ranking of the match by the corresponding “then” image when cued by a “now” image.*

4 Matching on Shape

We defined the shape vector of a face as the vector of (at present) 59 control points used in distorting the face image to the average shape. We describe here recognition results, similar to those above, matching on an invariant form of the shape vector, in which the effects of position and isotropic scale change have been removed. In principle we would expect to remove at least one other rotation parameter, to ensure that the resulting face was level, and perhaps a second, so the face is looking straight ahead, but in practice neither of these proved necessary for our images at present.

One way to remove position and scale parameters is to regard them as the only free parameters in a model of the face using all 59 control points. A fixed standard model is then chosen — typically the average shape again, and the fit between this model and the actual face shape calculated using euclidean distance between corresponding control points. A least squares minimisation of this error then provides “best fit” values of the free parameters, and the resulting “normalised” shape vector is one candidate on which to perform matching.

There are obvious problems with this approach, and we choose to first concentrate on those of the 59 points which are both “significant” and reliably located. Thus we generate a model consisting of only five points, obtained from the average of the 6 left eye points, the average of the 6 right eye points, the average of the 4 points at the end of the nose, the average of the 2 points at the ends of the mouth; and the point in the middle of the chin. We then perform the

least squares minimisation described above on this simple “eyes, nose, mouth and chin” model to remove three parameters associated with position and scale. A reduced shape vector was is a point in \mathbf{R}^7 , and the matches are ranked using euclidean distance in this space. As before, we report how well each target face matched the corresponding cue; the resulting ranks are given in Table 2.

Cue shape	Ian-now	Ken-now	Harry-now
Shape match	5	4	30

Table 2: *Matching faces based on their configuration vector. The table gives the rank, from a pool of 100 faces, of the match between the cue face and the corresponding target.*

We regard these results as preliminary and are currently exploring shape matching methods using explicit invariants, such as the ratio of the distance between the eyes to the mouth width. Such invariants, when combined with psychological ratings, have already been used successfully for recognition on fairly large sets of faces [17]. An important point is that matching is taking place on a very different characteristic from that contained in the texture vector, and that faces confused on one criterion are not confused on another; specifically, no face in our pool matches the cue better than the corresponding target on both shape and texture.

5 Principal Component Coding of Faces

We have presented our methodology as simple template matching, but in practice, the size of a texture vector (perhaps 30Kbytes for a 256×256 image) makes this unrealistically slow for large pools. The problem can be avoided using principal component analysis. As a first step, we choose a fixed set of shape-free faces, or an initial *ensemble*, and approximate all other faces as linear combinations of these. In the work we describe here, a total of 150 shape-free faces were first created. Of these, the three images destined to be targets were put directly in the pool, together with another 97, chosen using a random number generator. The remaining 50 images then became our initial ensemble, with which to represent all our other faces.

Our gain in efficiency comes from coding all shape-free faces in terms of the faces in the ensemble. Rather than code directly, we apply a principal component analysis to the shape-free faces in the ensemble. More precisely, we first obtain the mean image of the ensemble, and then the principal components, the eigenvectors of the covariance matrix, built from the deviations from the mean of each face in the ensemble. This gives a new basis of the subspace spanned by the ensemble; we refer to the basis elements (not themselves faces) as eigenfaces because of the way they are derived. The first six eigenfaces are shown in Fig. 4.

A shape-free face, even one not in the initial ensemble, can be approximated as a linear combination of these eigenfaces. The weights used in this sum, a total of 50 bytes, provide the succinct *eigenface representation* (cf [13], [22],

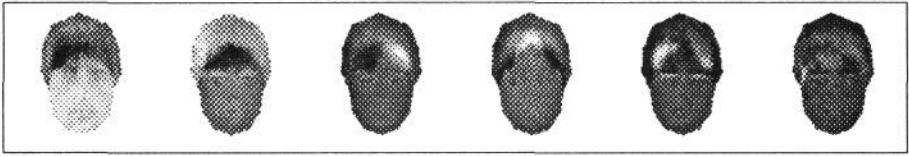


Figure 4: *The first six eigenfaces. Shape-free faces are represented as a linear combination (of typically 20) of these.*

[6]) which we use for matching. Note also that the errors in the approximation process are available. The whole process is illustrated in Fig. 6. We display the corresponding weighted sum of eigenfaces in the third image of Fig. 6. The fourth image then recombines the eigenface representation and the shape vector, to give the version of the original face that our representation effectively stores. A resemblance between the original and the reconstruction makes our ability to match using the reduced codes (see Table 3) more plausible.

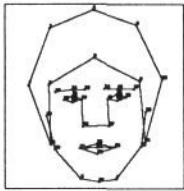


Figure 5: *The set of 40 control points found by FindFace.*



Figure 6: *The original is first distorted to a shape-free face to give the second image. This is approximated to give the third image; the distortion is then inverted to get the final reconstruction.*

We can both reduce the size of the representation still further, and improve its utility, by using only the most significant eigenfaces. The eigenfaces are initially chosen so that those corresponding to eigenvalues of large magnitude are good at discriminating between ensemble members while the bottom few code similarity. We thus choose to ignore the “unimportant” eigenfaces. There is significant debate on how many eigenvectors to take (eg [11], page 93); rather than keeping enough to capture 95% of the total variance, we first rescale the eigenvalues to have product 1, and discard those eigenfaces with corresponding rescaled eigenvalues less than 1. This will typically select 20 out of 50 eigenfaces, and in the process, capture over 80% of the variance in the ensemble.

We present results which are comparable to those in Table 1, using a pool of 100 faces as described above, and again using as cue each of the images shown in Fig. 2. All the images were first transformed to their shape-free state and then projected onto the subspace spanned by the most significant 20 eigenfaces obtained from the (totally disjoint) ensemble of 50 faces chosen above. The dot product between cue and each image in the pool was calculated, and the magnitude used as (naive) notion of similarity to rank the images in the pool as matches for the corresponding cue image. Table 3 gives these rankings.

Again we also give results when the portion of the image to be first coded,

and then matched, is restricted by excluding the hair. To do so, the ensemble of 50 faces used to generate the eigenfaces was also restricted in this way; and in this case, after normalising the eigenvalues, we were left with 21 eigenfaces (rather than 20) of magnitude at least 1, which were used for coding. The results seem significantly better than those in Table 1, although each image is now represented with approximately 20 bytes.

Cue face	Ian-now	Ken-now	Harry-now
Complete Face — top 20 components	1	27	10
Ignoring Hair — top 21 components	1	1	8

Table 3: *Matching shape-free faces in the Reduced Eigenface Representation. The rank, from a pool of 100 faces, of the match between a cue face and the corresponding target is shown.*

For comparison, Turk and Pentland [22], work with a pool containing many images of 16 individuals, build an ensemble with a different image of each individual, and extract 7 eigenfaces. They then classify each face in the pool as one of the sixteen individuals. However the registration, which we regard as vital, is only implicit in their work, and is only done with global position and scaling parameters, rather than with a full anticarcature; indeed size and position are variables they seek to recognise across.

One can view the passage from the full shape-free face, to the reduced representation, in terms of only a few significant eigenfaces: either as a useful approximation to the “true” full template, in which the loss of accuracy is compensated for by the speed with which matching is performed; or as an improvement over crude template matching, in which, by passing to a few codes chosen for their ability to describe variability between faces, we have obtained the ability to generalise from the particular image used as a cue. We believe the evidence above supports the latter view; and that the generalisation occurring is similar to that which can occur in a neural net. We note also, again in comparison with net-based systems, that no knowledge of the cue or target is available to the coding system, and no training is needed to extract “suitable” codes.

Finally we note that even our initial description of recognition can be phrased in terms of principal component analysis: using the full encoding, with an ensemble which co-incides with our pool, is equivalent (up to a fixed choice of weights) to the template matching we described in Section 3.

6 Theoretical Discussion and Conclusions

We now place our coding scheme in a theoretical context. Our two stages of coding can be considered as a two stage description of the underlying geometry, (see Craw and Cameron [6]) in which the “face manifold” is first subjected to non-linear perturbations, yielding a linear structure, on which familiar linear operations can be applied [3]. This need for linearity may explain why the

codes used for recognition also arise when trying to perform realistic merges (or linear averages) between two faces.

We have explicitly described processing needed to perform recognition; it may be that similar processing occurs within a neural net dedicated to face recognition. The first phase, in which features are identified, and the image is linearised to a shape-free face, is a familiar non-linear warping step, frequently observed and well understood (eg [10], page 235). Since we follow this by principal component analysis, essentially the function performed by linear neural nets [15], the overall functioning of our proposed method may be sufficiently similar to that of a neural net to provide insight into their functioning. Learning the need for the non-linear warp will require much training, and it is here we obtain a computational advantage: at present by essentially performing this time-consuming operation manually; and in principle by applying FindFace, whose individual modules can be tuned for performance.

We claim to have demonstrated a useful recognition performance even when there is a much more significant difference between target and cue than is usual. Further, we have the possibility of doing this completely automatically. Although our main interest is in the principal component analysis based texture recognition, our experiments with simple template matching suggest that subsequent results are not an artifact, and that the concept of a shape-free face is useful. Our results on shape-based recognition are more tentative, although it is a much more familiar method; its interest is in providing a relatively independent measure of recognition, to combine with our texture measure yielding a single measure which is more discriminating than either.

We believe our coding scheme has promise, but wish to know more of its limitations. In part these arise from inaccurate control point locations, in part from the approximation inherent in principal component analysis, and in part from the sheer variability of the human face on different occasions; the relative importance of these errors is not yet clear. Template matching methods are notoriously dependent on lighting, and we have done little to explore this. One approach incorporates so much lighting variation in the ensemble that no particular condition is accurately encoded. Control of lighting for the pool images, as used for our tests, provides a way to finesse this difficulty. Another aspect to explore is the choice of initial ensemble; it should be selected to provide good representations of the types of faces being coded, rather than randomly. We noted in Section 4 the possibility of incorporating a limited view-invariance in our recognition process; by using a number of ensembles, each tuned to a specific angle of view, we may provide a much greater degree of view invariance.

References

- [1] R. J. Baron. Mechanisms of human facial recognition. *International Journal of Man-Machine Studies*, 15:137-178, 1981.
- [2] P. J. Benson and D. I. Perrett. Perception and recognition of photographic quality facial caricatures: Implications for the recognition of natural images. *European Journal of Cognitive Psychology*, 3(1):105-135, 1991.
- [3] V. Bruce, A. M. Burton, and I. Craw. Modelling face recognition. *Philosophical Transactions of the Royal Society of London, Series B*, 335:121-128, 1992.

- [4] R. Brunelli and T. Poggio. HyberBF networks for gender classification. Preprint, 1991.
- [5] G. W. Cottrell and M. Fleming. Face recognition using unsupervised feature extraction. In *Proceedings of the International Neural Net Conference*, pages 322–325. Dordrecht Kluwer, 1990.
- [6] I. Craw and P. Cameron. Parameterising images for recognition and reconstruction. In P. Mowforth, editor, *British Machine Vision Conference 1991*, pages 367–370, London, 1991. Springer Verlag.
- [7] I. Craw, D. Tock, and A. Bennett. Finding face features. In G. Sandini, editor, *Proceedings of ECCV-92*, number 588 in Lecture Notes on Computing Science, pages ??–?? Springer-Verlag, 1992.
- [8] R. Gallery and T. I. P. Trew. An architecture for face classification. In *Colloquium: Machine Storage and Recognition of Faces. IEE Digest 017*, 1992.
- [9] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computing*. Computation and Neural Systems series. Santa Fe Institute and Addison Wesley, 1991.
- [10] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
- [11] T. Kanade. *Computer Recognition of Human Faces*, volume 47 of *Interdisciplinary Systems Research*. Birkhäuser, Basel, Stuttgart, 1977.
- [12] M. Kirby and L. Sirovich. Application of the karhunen-loève procedure for the characterisation of human faces. *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.
- [13] T. Kohonen, E. Oja, and P. Lehtiö. Storage and processing of information in distributed associative memory systems. In G. Hinton and J. Anderson, editors, *Parallel models of associative memory*, chapter 4. Erlbaum, Hillsdale N.J., 1981.
- [14] R. Linsker. From basic network principles to neural architecture: Emergence of orientation columns. *Proceedings of the National Academy of Sciences*, 83:8779–8783, 1986.
- [15] R. M. Rickman and J. Stonham. Coding facial images for database retrieval using a self organising neural network. In *Colloquium: Machine Storage and Recognition of Faces. IEE Digest 017*, 1992.
- [16] J. W. Shepherd. An interactive computer system for retrieving faces. In H. D. Ellis, M. A. Jeeves, F. Newcombe, and A. Young, editors, *Aspects of Face Processing*, chapter 10, pages 398–409. Martinus Nijhoff, Dordrecht, 1986. NATO ASI Series D: Behavioural and Social Sciences - No. 28.
- [17] R. B. Starkey and I. Aleksander. Facial recognition for police purposes using computer graphics and neural networks. In *Colloquium: Electronic Images and Image Processing in Forensic Science. IEE Digest 087*, 1990.
- [18] T. Stonham. Practical face recognition and verification with WISARD. In H. Ellis, M. Jeeves, F. Newcome, and A. Young, editors, *Aspects of Face Processing*, pages 426–441. Martinus Nijhoff, Dordrecht, 1986.
- [19] K. Sutherland, D. Rensham, and P. B. Denyer. A novel automatic face recognition algorithm employing vector quantization. In *Colloquium: Machine Storage and Recognition of Faces. IEE Digest 017*, 1992.
- [20] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.