Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 26 (2008) 382–396
www.elsevier.com/locate/imavis

Three-dimensional face recognition using combinations of surface feature map subspace components

Thomas Heseltine *, Nick Pears, Jim Austin

Advanced Computer Architecture Group, Department of Computer Science, The University of York, Heslington, York, YO10 5DD, United Kingdom

Received 24 February 2005; received in revised form 22 May 2006; accepted 8 December 2006

Abstract

In this paper, we show the effect of using a variety of facial surface feature maps within the Fishersurface technique, which uses linear discriminant analysis, and suggest a method of identifying and extracting useful qualities offered by each surface feature map. Combining these multi-feature subspace components into a unified surface subspace, we create a three-dimensional face recognition system producing significantly lower error rates than individual surface feature map systems tested on the same data. We evaluate systems by performing up to 1,079,715 verification operations on a large test set of 3D face models. Results are presented in the form of false acceptance and false rejection rates, generated by varying a decision threshold applied to a distance metric in surface space.

Keywords: Face recognition; Three-dimensional; Depth map; Range data; Multi-feature; Subspace combinations

1. Introduction

Despite significant advances in face recognition technology, it has yet to achieve levels of accuracy required for many commercial and industrial applications. The high error rates stem from well-known sub-problems. Variation in lighting, facial expression and orientation all significantly increase error rates. In an attempt to address these issues, research has begun to focus on the use of three-dimensional face models, motivated by three main factors. First, relying on geometric shape, rather than colour and texture information, systems become invariant to lighting conditions. Second, the ability to rotate a facial structure in three-dimensional space, allowing for compensation of variations in pose, aids those methods requiring alignment prior to recognition. Third, the additional depth information in the facial surface structure, not directly available from two-dimensional images, provides supplementary cues for recognition.

In this paper, we expand on our previous research [1] involving the use of facial surface data, derived from 3D face models (generated using a stereo vision 3D camera), as a substitute for the more familiar two-dimensional images. To date numerous investigations have shown that three-dimensional structure can be used to aid recognition. Zhao and Chellappa [2] use a generic 3D face model to normalise facial orientation and lighting direction in two-dimensional images, increasing recognition accuracy from approximately 81% (correct match within rank of 25) to 100%. Similar results are witnessed in the face recognition vendor test [3], showing that pose correction using Romdhani et al's technique [4] reduces error rates when applied to the FERET database. In this work parameters affecting the shape, orientation and texture of the morphable face model are varied and this is then projected onto a 2D plane for comparison with the two-dimensional image. The iteration continues until the difference between the projected image and original image is minimised. The orientation parameters can then be ignored, while the shape and texture parameters are taken as features for identification, resulting in 82.6% correct identifications on a test set of 68

* Corresponding author. Tel.: +44 1904 432722; fax: +44 1904 432767.
E-mail address: tomh@cs.york.ac.uk (T. Heseltine).

0262-8856/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.imavis.2006.12.008

people. However, due to its high computational complexity, this method may not be suitable for applications that require an immediate response, such as secure site access, time and attendance systems and surveillance. Blanz et al. [5] take a comparable approach, using a morphable face model to aid in identification of two-dimensional images. The method is initiated with an estimate of lighting direction and a generic three-dimensional facial surface.

Although the methods described above do show that knowledge of three-dimensional face shape can aid normalisation for two-dimensional face recognition systems, none of the methods mentioned so far use actual three-dimensional geometric structure to perform recognition. Although Beumier and Acheroy [6,7] do make direct use of such information, when testing various methods of matching 3D face models, few were successful. Curvature analysis proved ineffective, and feature extraction was not robust enough to provide accurate recognition. However, Beumier and Acheroy were able to achieve reasonable error rates using curvature values of vertical surface profiles. Verification tests carried out on a database of 30 people produced equal error rates (EER) between 7.25% and 9.0%. Hesher et al. [8] test a different method, using principal component analysis (PCA) of depth maps and euclidean distance to perform identification with 94% accuracy on 37 face models (when trained on the gallery set).

Further investigation into this approach is carried out in our earlier work [9]. Again applying PCA to facial surface depth maps and testing on a database of 290 face models that present typical difficulties for recognition, we have shown that the EER can vary from 33.1% to 17.8%, depending on the distance metric used in the surface space. We have also demonstrated the effects of a variety of 3D surface feature maps, such as spatial derivatives and curvature values, which reduce the EER from 19.1% using the initial depth map representation and a Euclidean distance metric, down to 12.7% using the best surface feature map in conjunction with a cosine distance metric.

We have also applied fisher's linear discriminant analysis (LDA) to the same facial surface feature maps [1]. We term this approach the Fishersurface method, to distinguish it from the Belhumeur et al's two-dimensional fisherface method [10]. However, the focus of our previous research has been to identify discriminating surface feature maps (such as depth, gradient or curvature maps), with little regard for the advantages offered by each individual representation. Here, we suggest that specific surface features maps may be specifically suited to specific capture conditions or certain facial characteristics, despite the feature map being poor overall for recognition. For example, curvature feature maps may aid recognition by making the system more robust to inaccuracies in 3D orientation, yet may also be highly sensitive to noise. Another type of feature map may enhance nose shape, but lose information regarding jaw structure.

In this paper, we use LDA to determine a maximally discriminating sub-space for each of seventeen different surface feature maps. Within these subspaces, we examine the relative discriminating power of the components within each of those subspaces. We then propose a means of combining individual components from all of these subspaces into a single unified space, in which face discrimination and hence recognition accuracy is improved, when compared with any sub-space based on a single feature surface map.

Pentland et al. [11] have previously examined the benefit of using multiple eigenspaces, in which specialist subspaces were constructed for various facial orientations, from which cumulative match scores were able to reduce error rates. Our approach differs in that we extract and combine individual dimensions, creating a single unified surface space. For example, we may take the first two dimensions of a subspace created from LDA of depth maps and combine these with the fifth, ninth and tenth dimensions taken from a subspace of a curvature feature map. This method has already been proven a success when applied to two-dimensional images [12] and we conjecture that similar improvements will be witnessed when applied to three-dimensional systems.

The paper is structured as follows: in Section 2, we describe how we have captured our data and built up the University of York 3D face database. Section 3 describes pose normalisation, such that all faces are orientated to a fronto-parallel pose. Section 4 then describes how we take pose-normalised data and generate raw depth maps and other (derived) 3D surface feature maps, such as gradient and curvature maps. Since we have multiple examples of each 3D face in our database, we can project our data to a maximally discriminating sub-space using LDA, for each individual surface feature type. We have called LDA applied to 3D faces the ''Fishersurface Method'' and Section 5 describes this in detail. Section 6 then applies LDA to each individual surface feature map and examines the relative performance of a range of single feature face recognition systems. Section 7 then describes how we combine components (individual dimensions from individual, single feature sub-spaces) to build a multi-feature sub-space that is more effective for face recognition than any single feature system alone. Section 8 presents our evaluation procedure and our results, while a final section is used for conclusions.

2. Data capture and the University of York 3D face database

In this section we discuss the nature of the 3D face data being used in these experiments, and how this is used to generate 3D surface feature maps.

The 3D face data (models) are captured and generated in sub-second processing time from a single shot with a 3D camera, which operates on the basis of stereo disparity of a high-density projected light pattern. The unit consists of two cameras, from which the images are used to produce a three-dimensional point cloud of the facial surface. A third camera is used to capture texture information, which is subsequently mapped onto the 3D model, as seen in Fig. 1 (bottom right). A calibration procedure is performed

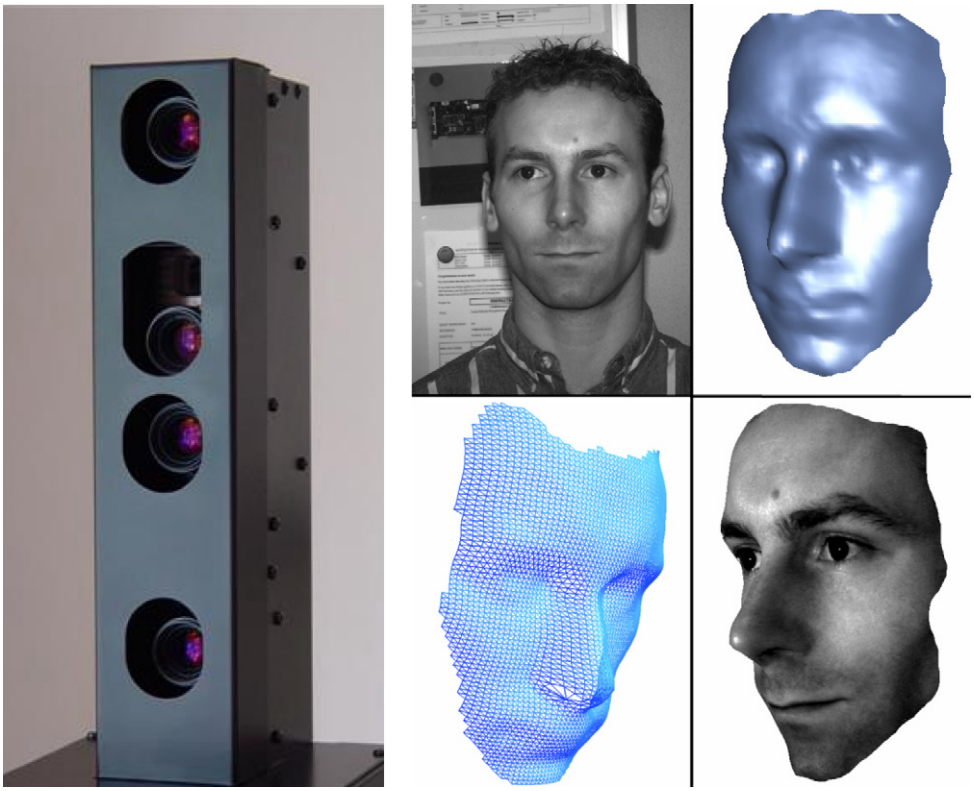

Fig. 1. 3D capture camera (left) and example 3D face models (right).

to determine the intrinsic and extrinsic parameters of the stereo cameras. The 3D facial surfaces are generated using standard 3D reconstruction techniques and output in Wavefront's OBJ file format.

In its simplest form, the 3D face model is a set of points in three-dimensional space, with each point lying on some object surface. This means that the point cloud actually describes the nearest visible surface to the 3D camera and any areas that are occluded from one of the stereo cameras will result in gaps in the point cloud, as can be seen in Fig. 2 around the nostril and ear regions. (This is effectively a 2.5D representation from which we can derive a depth map.)

In addition to this point cloud data the OBJ file format also includes polygon information. Each polygon is defined as a reference to three neighbouring points (or vertices), hence describing a three-dimensional triangular mesh. This data allows for production of smooth polygonal visualisations (and ultimately full texture mapping) as well as wire-mesh representations, which is useful for navigating between locally connected vertices in surface processing techniques.

Note that the resolution of 3D face models cannot be clearly defined as with 2D images that have a defined and consistent number of pixels in each image, across a spatially array. Here, we specify the resolution of the 3D face models used in these experiments simply by the number of points on the surface of the face, which typically number in the range of 5000–6000, but this figure should only be taken as a general guide to point resolution.

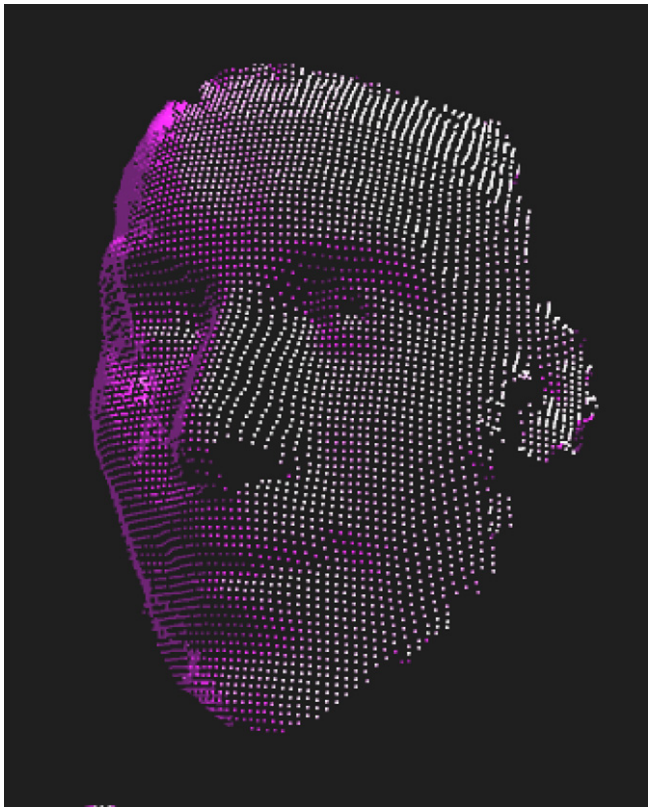

Fig. 2. Point cloud of three-dimensional face surface.

This capture process has been used to populate a large 3D face database owned by the University of York (UK) and recently made available as part of an ongoing project to provide a publicly available database of 3D face models [13]. The standard data set contains 15 capture conditions per person, a subset of which can be seen in Fig. 3. The database now consists of over 5000 models of over 350 people, making it one of the largest 3D face databases currently available.

For the purpose of these experiments we select a sample of 1770 face models (280 people) captured under the conditions in Fig. 3. During data acquisition no effort was made to control lighting conditions and the faces have a variety of facial expressions. In order to generate face models at various head orientations, subjects were asked to face reference points positioned roughly 45° above and below the camera, but no effort was made to enforce precise orientation.

3. Pose normalisation

Many face recognition algorithms require some method of alignment prior to performing recognition, including two-dimensional PCA and LDA approaches. Similarly, three-dimensional face recognition requires alignment of the 3D surfaces before recognition takes place. The orientation must take place across all six degrees of freedom (three directions of rotation and three directions of translation), which may be achieved by aligning three points on the 3D face surface. However, for this to be successful, we must consistently localise these three points regardless of the orientation of the face at the time of capture. This has proven to be a particularly difficult task and forms a significant part of our current research in 3D face recognition. In the work presented here, we have used a simple 3D face alignment algorithm, which has proven to be robust under the following assumptions:

- The tip of the nose is visible.
- The nose tip is the most protruding object on the 3D surface.
- The face is within 45 degrees of pan and tilt angle relative to fronto-parallel pose.

We apply the 3D orientation normalisation algorithm in a similar manner to typical 2D image alignment procedures. After localising facial landmarks, we translate and rotate all face models into a front-facing orientation prior to any training, verification or identification procedures. In 2D systems, localising the eye centres allows for image alignment. In terms of colour and texture, the eyes are well-defined, unique areas of the face with precise and easily detected centres (the pupils), but with the absence of texture information (when using purely geometric information) this is not the case. As seen in Fig. 4, when texture information is not available, pinpointing the centre of the eyes is particularly difficult, even for the human vision system.

Theory suggests that we require three points on the facial surface to align a 3D model. However, we are faced with the problem that there are few facial landmarks that are easily detected when using surface shape alone. Therefore, we have developed an algorithm that uses many more points, creating a more robust solution, relying on multiple

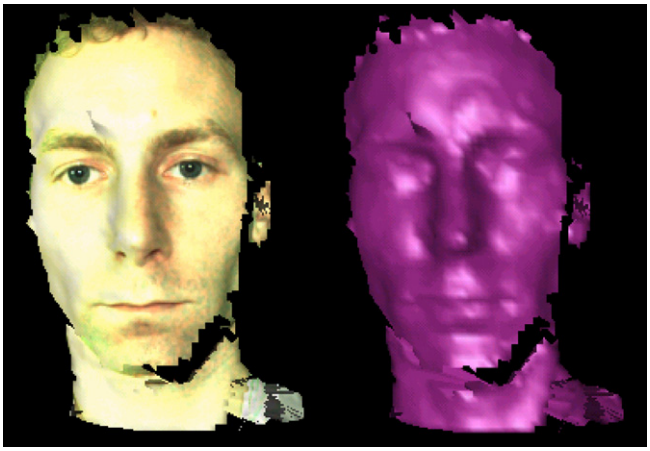

Fig. 4. 3D facial surface data viewed with and without texture mapping.

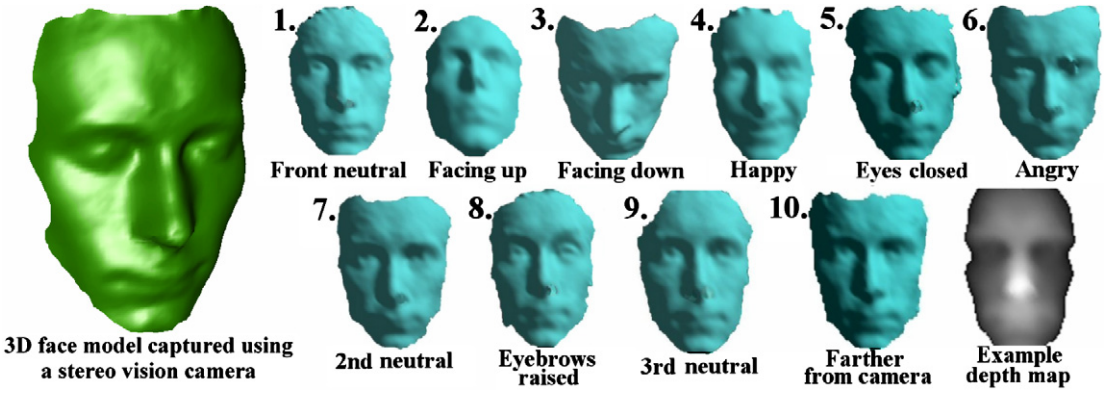

Fig. 3. Example face models taken from the University of York 3D face database.

```
Centre of rotation = mean vertex
For x-rotation from -45 to 45, step 5:
      For y rotation from -45 to 45 step 5:
                  Flag most forward vertex
Nose tip = vertex with most flags
```

Fig. 5. Nose tip localisation algorithm.

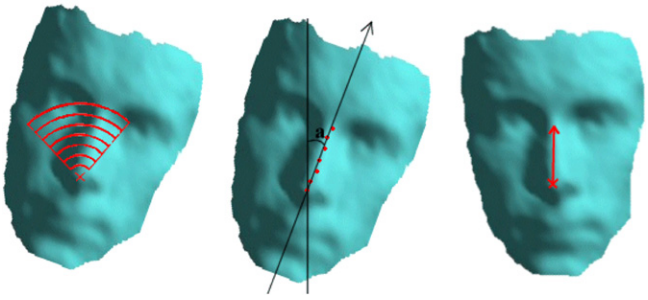

Fig. 6. 3D orientation normalisation about the z-axis.

redundancy and majority voting. The algorithm consists of four stages as follows:

(i) *Nose tip detection*. The most easily located facial feature is the nose tip, and it is for this reason that we begin orientation normalisation by locating this feature. All subsequent procedures rely on successful detection of the nose tip. The approach we take is to identify the most protruding point on the surface. If the head is in a fronto-parallel orientation, the nose can be identified as the most forward vertex (the vertex with the smallest z coordinate). However, as we do not know which way the subject will be facing, we must iteratively rotate the surface about the x (tilt) and y (pan) axis, and build a histogram populated by the most forward vertex on each iteration. Providing each increment in rotation angle is sufficiently small, the result is that the nose tip has the smallest z coordinate at a higher frequency than any other vertex (Fig. 5).Having located the nose tip, we translate the 3D surface such that the nose tip is located at the origin of the coordinate space, thus normalising the x, y and z position of the face in 3D space to the resolution of the point cloud. Note that it is possible to use interpolating bicubic surface patches, such as Hermite patches, to improve the resolution of nose tip localisation and, although not used here, this is one of several methods being evaluated in our ongoing work in 3D face alignment.

(ii) *Roll correction*. We then rotate the 3D model about the z-axis, such that the face becomes vertical, thus normalising rotation about the z-axis. This requires that we locate the bridge of the nose by searching for the most forward vertices within a 90-degree arc of concentric radii above the nose tip, as shown on the left of Fig. 6. This provides a set of points (one on each radii) along the bridge of the nose, from which we take the least squares line of best fit as a vector indicating the nose bridge direction. We then rotate the 3D model about the z-axis, such that the nose bridge vector becomes vertical in the x–y plane, thus normalising rotation about the z-axis.

(iii) *Tilt correction*. Rotating the 3D model, such that the forehead is directly above the nose tip, normalises tilt orientation about the x-axis. Initially, one may suggest that the bridge of the nose could also be used to normalise rotation about the x-axis (by ensuring that a point on the nose bridge is located directly above the nose tip). However, we have found this method to produce imprecise alignment, as just a small mis-localisation along the nose bridge can result in large discrepancies in the degree of corrective rotation applied. A much more suitable point to use in normalising tilt about the x-axis would be located on the forehead, due to the relatively flat surface structure (and hence little impact through mis-location). Therefore, the next step is to locate a point on the facial surface intersecting with the plane $x = 0$, at a distance F (typically 90 mm) from the nose tip.

(iv) *Pan correction*. The final step in the alignment procedure is to correct head pan, by rotating about the y-axis to achieve a fronto-parallel alignment. This is done by locating points of intersection between the facial surface, an arbitrary horizontal plane and a vertical cylinder, centred about the nose tip. For a given horizontal plane and cylinder of radius W there will be two points of intersection: one on the left side of the face and one on the right, as shown in Fig. 7 (left). By adjusting the y-coordinate of the horizontal plane, we produce a set of intersection point pairs on the facial surface (Fig. 7, right).

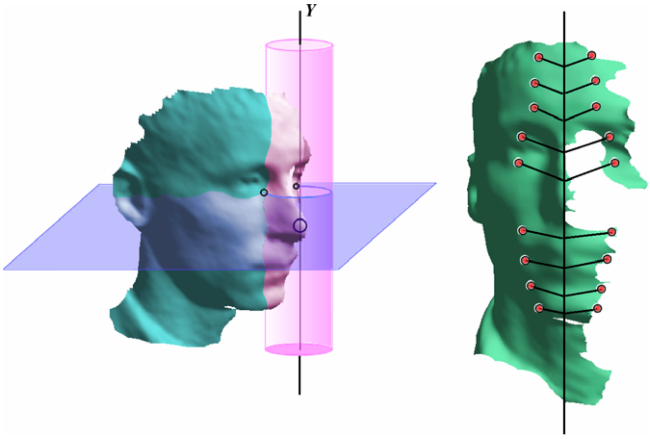

Fig. 7. 3D orientation normalisation about the y-axis.

To achieve a fronto-parallel alignment, the left and right points of each pair should have the same z-coordinates. We calculate the required angle of rotation about the y-axis to achieve this balance and then repeat the calculation for a set of even spaced horizontal planes. If no point of intersection exists (due to an incomplete 3D surface), then that horizontal plane is ignored. If few point pairs are detected then the radius of the cylinder can be adjusted and the process repeated. This is often necessary if the head is particularly small (i.e. a child's head), meaning that the face is wholly contained within the cylinder.

The final degree of rotation (about the y-axis) is calculated as the average of all corrective angles for the point pairs. This averaging method helps to compensate for noise, facial distortion (due to expression) or small non-face protrusions (headwear).

4. Generation of depth maps and other 3D surface feature maps

Once all of the 3D face data is aligned to a common orientation using the procedures described in the above section, we generate a depth map for each face model. A depth map is analogous in structure to a standard image, in that a regular rectangular array of values describes the data. Here, however, those values represent depth rather than image intensity and 'pixels' are regions on the x–y plane in 3D space rather than in image space. Furthermore, when the 3D point cloud is projected into a depth map, all of the data is aligned, due to the alignment process in 3D space. The image on the right of Fig. 8 shows the depth map after orientation normalisation has been applied. As with all our depth maps, it is 60 pixels wide by 90 pixels high and it is rendered such that points near the camera are bright and points farther from the camera are dark.

Once all of the data in our database has been put into (aligned) depth map form, it is easy to see how popular appearance based 2D face recognition approaches, which project the data into a sub-space can be applied. The most common approaches are Eigenface (PCA) and, for datasets which contain multiple examples of the same person, Fisherface (LDA).

It is well known that the use of image processing can significantly reduce error rates of pattern matching applications for standard 2D images by removing unwanted effects caused by environmental capture conditions, as demonstrated in previous work experimenting with 2D face recognition methods [12,10,13]. Much of this environmental influence is not present in the 3D face models, but depth map processing may still aid recognition by making distinguishing features more explicit and reducing noise.

In this work, we have processed raw depth maps in a number of different ways to produce a set of surface feature maps, which include features such as gradient and curvature, expressed over the same rectangular array as the raw depth maps themselves. The various feature maps expose relationships between 3D vertices otherwise not taken into account when LDA is applied directly to depth maps. It is also thought that gradient maps may be more robust to translations along the z-axis and curvature maps more resilient to small inaccuracies in orientation, although also more susceptible to noise. The full list of surface feature maps are as follows (pictured in order, left to right in Fig. 9):

- Depth map.
- Horizontal gradient map (at two scales).
- Vertical gradient map (at two scales).
- Laplacian map (second order derivative).
- Sobel gradient map (in x and y dimensions).
- Sobel magnitude map.
- Curvature map (in x and y dimensions).
- Curvature magnitude map.
- Curvature type map.
- Convexity map (minimum curvature).
- Concavity map (maximum curvature).
- Absolute minimum curvature map.
- Absolute maximum curvature map.

The detail of how these surface feature maps are generated is given in Appendix A. These methods are applied to

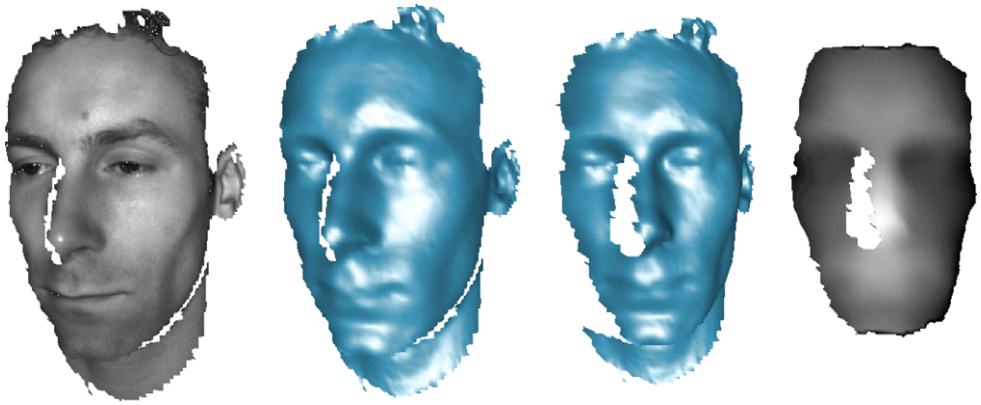

Fig. 8. Original 3D face model (left two), orientation normalised 3D face model and depth map representation (right two).

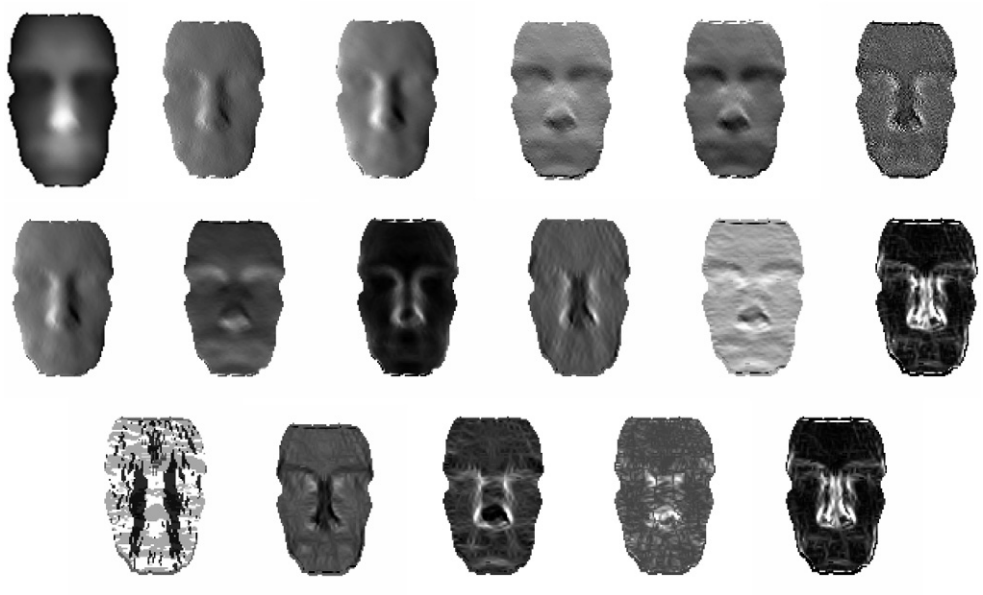

Fig. 9. The seventeen surface feature maps used to create seventeen subspaces, later combined into a single unified face recognition system.

depth map images prior to any further analysis in either the training or test procedures. Typically, processing algorithms are applied to the training and test sets of depth maps as a batch process and the resulting images stored on disk, ready for eigenvector analysis or face subspace projection later, such that a separate surface space is generated for each surface feature map and hence creates a separate face recognition system. The method we use to implement a face recognition system is subspace generation using linear discriminant analysis, which we describe in the next section.

5. The Fishersurface method

In this section we provide details of the Fishersurface method of face recognition. We use the term Fishersurface method to describe the application of LDA (linear discriminant analysis) to surface feature maps of 3D face models in order to produce subspace projection matrices, as with Belhumier et al's fisherface approach [10]. The method can be applied to a variety of surface feature maps, creating a series of projection matrices, each corresponding to a specific type of 3D surface feature. Taking advantage of 'within-class' and 'between-class' information, we maximise the ratio of between-class to within-class separation. Thus creating a subspace in which the variation between face models of the same person is minimal, relative to the much greater deviation between images of different people.

To accomplish this we define a training set τ, of surface feature maps in vector form, shown in Eq. (1). This training set is populated with samples of one specific type of feature map and partitioned into c classes (where c equals the number of different people present in the training set). Each class X_n is comprised of a number of surface feature maps $\boldsymbol{\Gamma}_{ni}$, such that all feature maps in a single class are of the same person and no one person is present in multiple classes.

$$\begin{aligned} &\tau = \{X_1, X_2, \ldots, X_C\} \\ &\text{where } X_n = \{\boldsymbol{\Gamma}_{n1}, \boldsymbol{\Gamma}_{n2}, \boldsymbol{\Gamma}_{n3}, \ldots\} \end{aligned} \tag{1}$$

For each type of feature map described in Appendix A, we create a separate training set containing the same subjects but represented by a different type of surface feature map. For clarity, we now continue to describe the Fishersurface method as applied to a single feature map, bearing in mind this process will be repeated for each feature map type and later combined as described in Section 7.

From τ we calculate the average feature map Ψ_n for each class X_n, as well as the average of all feature maps Ψ, using the formulae shown in Eq. (2).

$$\boldsymbol{\Psi} = \frac{\sum_{n=1}^{C}\sum_{i=1}^{|\mathbf{X}_n|}\boldsymbol{\Gamma}_{ni}}{\sum_{m=1}^{C}|\mathbf{X}_m|} \qquad \boldsymbol{\Psi}_n = \frac{1}{|\mathbf{X}_n|}\sum_{i=1}^{|\mathbf{X}_n|}\boldsymbol{\Gamma}_{ni} \tag{2}$$

From these average feature maps we are then able to compute three scatter matrices describing the variance of facial surface structure throughout the training set. First, we compute the between-class scatter matrix $\boldsymbol{S}_B$, representing the natural variance in facial structure from one person to another, using Eq. (3). It is this variance between different people that we wish to accentuate within our final surface space.

$$\boldsymbol{S}_B = \sum_{n=1}^{C}|\boldsymbol{X}_n|(\boldsymbol{\Psi}_n - \boldsymbol{\Psi})(\boldsymbol{\Psi}_n - \boldsymbol{\Psi})^T \tag{3}$$

The next scatter matrix $\boldsymbol{S}_w$, is calculated using Eq. (4). This matrix describes the variance due to other influences, such as facial expression, minor discrepancies in alignment and other noise that may occur between different facial surfaces

acquired from the same person, which we hope to suppress in the final surface space produced.

$$\boldsymbol{S}_W = \sum_{n=1}^{C} \sum_{i=1}^{|\boldsymbol{X}_n|} (\boldsymbol{\Gamma}_{ni} - \boldsymbol{\Psi}_n)(\boldsymbol{\Gamma}_{ni} - \boldsymbol{\Psi}_n)^T \tag{4}$$

If we were able to ensure that $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ were non-singular, we could omit this next step in the Fishersurface method and perform LDA directly, using the ratio of these two matrices. However, with little training data relative to the dimensionality of the vectors, we must firstly reduce the dimensionality of the two scatter matrices using PCA. This is done by computing the eigenvectors of a third scatter matrix $\boldsymbol{S}_T$, as shown in Eq. (5), and taking the top 250 (the total number of feature maps minus c) principal components to produce a projection matrix $\boldsymbol{U}_{\text{pca}}$. $\boldsymbol{S}_T$ describes the variance across the entire surface space for the given type of feature map used.

$$\begin{aligned} \boldsymbol{U}_{\text{pca}} &= \arg\max_{U} \left(|\boldsymbol{U}^T \boldsymbol{S}_T \boldsymbol{U}| \right) \\ \text{where } \boldsymbol{S}_T &= \sum_{n=1}^{C} \sum_{i=1}^{|\boldsymbol{X}_n|} (\boldsymbol{\Gamma}_{ni} - \boldsymbol{\Psi})(\boldsymbol{\Gamma}_{ni} - \boldsymbol{\Psi})^T \end{aligned} \tag{5}$$

This projection matrix $\boldsymbol{U}_{\text{pca}}$ is ultimately used to reduce dimensionality of the within-class and between-class scatter matrices $\boldsymbol{S}_w$ and $\boldsymbol{S}_B$, ensuring they are non-singular before computing the top $c-1$ eigenvectors of the reduced scatter matrix ratio, $\boldsymbol{U}_{\text{fld}}$, as shown in Eq. (6).

$$\boldsymbol{U}_{\text{fld}} = \arg\max_{U} \left(\frac{\boldsymbol{U}^T \boldsymbol{U}_{\text{pca}}^T \boldsymbol{S}_B \boldsymbol{U}_{\text{pca}} \boldsymbol{U}}{\boldsymbol{U}^T \boldsymbol{U}_{\text{pca}}^T \boldsymbol{S}_W \boldsymbol{U}_{\text{pca}} \boldsymbol{U}} \right) \tag{6}$$

Finally, we produce the projection matrix $\boldsymbol{U}_{ff}$ in Eq. (7), such that we may project a surface feature map into a reduced space of $c-1$ dimensions using a single projection matrix. The resulting subspace maximises the ratio of between-class scatter (for all c classes) to within-class scatter (for each class X_n).

$$\boldsymbol{U}_{\text{ff}} = \boldsymbol{U}_{\text{fld}} \boldsymbol{U}_{\text{pca}} \tag{7}$$

Like the fisherface system [10], components of the projection matrix $\boldsymbol{U}_{\text{ff}}$ can be viewed as images, as shown in Fig. 10 for the (raw) depth map surface space.

Once surface space has been defined, we project a vector form of the facial surface $\boldsymbol{\Gamma}$ into the reduced surface space by a simple matrix multiplication, as shown in Eq. (8). The resultant vector $\boldsymbol{\Omega}^T = [\omega_1, \omega_2, \ldots, \omega_{c-1}]$ is taken as a 'face-key' representing the facial structure in reduced dimensionality space

$$\boldsymbol{\Omega} = (\boldsymbol{\Gamma} - \boldsymbol{\Psi})^T \boldsymbol{U}_{\text{ff}} \tag{8}$$

Face-keys may be compared using a variety of distance metrics. Here, we test two, the simple euclidean distance and the cosine distance as shown in Eq. (9), the results of which are compared in Sections 6 and 8.

$$d_{\text{euclidean}} = \|\boldsymbol{\Omega}_a - \boldsymbol{\Omega}_b\| \quad d_{\cos\text{ine}} = 1 - \frac{\boldsymbol{\Omega}_a^T \boldsymbol{\Omega}_b}{\|\boldsymbol{\Omega}_a\| \|\boldsymbol{\Omega}_b\|} \tag{9}$$

An acceptance (facial surfaces match) or rejection (surfaces do not match) is determined by applying a threshold to the distance calculated. Any comparison producing a distance value below the threshold is considered an acceptance. By adjusting this threshold value, one can change the balance between the number of false acceptances and false rejections, making the system either more secure or more tolerant to changing conditions. Secure site access systems would typically set the threshold such that false acceptances were significantly lower than false rejections: unwilling to tolerate intruders at the cost of inconvenient access denials. Surveillance systems on the other hand would require low false rejection rates to successfully identify people in a less controlled environment. By adjusting the threshold through the full range of possible distance values we are able to produce error curves that describe the operating characteristics across all threshold values, as presented in Section 8. In order to select a suitable threshold for a specific application, the FAR and FRR error curves would be plotted as a function of threshold and having chosen a suitable balance of FAR to FRR the required threshold value can be observed (the sensitivity of which is related to the gradient of the error curve).

Note that, in order to make a comparison of systems that is application neutral, we choose the equal error rate (EER) as the single comparative value for system performance. This is the misclassification error when FAR is equal to FRR, a lower value indicating a better system performance over the data test set.

6. Face recognition using individual surface feature maps

This work aims to apply the Fishersurface technique to individual surface feature maps in order to determine feature specific recognition performance. It then aims to com-

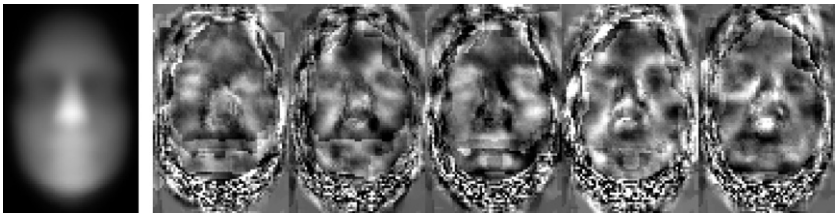

Fig. 10. The average surface (left) and first five Fishersurfaces (right), in which lighter and darker regions indicate greater positive and negative variance, respectively, while mid-grey pixels indicates zero variance.

bine a mixture of subspace components across all surface feature subspaces to give a recognition system with improved performance over the best single feature recognition system. We therefore have to be careful about using a common training and test data set for all experiments.

In order to do this, we take a training set of 300 depth maps (50 people), which is used to compute the scatter matrices described in the previous section. The remaining 1470 depth maps (230 people) are then separated into two disjoint sets of equal size (test set A and test set B). We use test set A to analyse the face-key variance throughout each surface feature subspace, calculate discriminant weightings and compute the optimum surface space combinations. This leaves set B as an unseen test set to evaluate the final combined system. Both training and test sets contain subjects of various race, age and gender and nobody is present in both the training and test sets.

First, we examine surface feature specific sub-spaces and the overall performance produced when specific surface feature maps are used within the Fishersurface method. We begin by testing the variety of surface feature maps on test set A and the range of error rates produced is shown in Fig. 11. Note that the use of the cosine distance metric is consistently better than the Euclidean distance metric. Clearly, within a feature subspace, faces seem to be better discriminated based on their angular separation rather than distance separation.

Figure 11 also clearly shows that the choice of surface feature map has a significant impact on the effectiveness of recognition, with horizontal gradient surface feature maps providing the lowest equal error rate (EER, the point at which false acceptance rate equals false rejection rate).

However, the superiority of the horizontal gradient surface feature map does not suggest that the vertical gradient and curvature feature maps are no use whatsoever. Although discriminatory information provided by these features may not be as robust and distinguishing, they may contain a degree of information not available in horizontal gradients and could therefore still make a positive contribution to a combined surface space. We measure the discriminating ability of surface space dimensions by applying Fisher's linear discriminant (FLD) (as used by Gordon [14]) to individual components (single dimensions) of each surface subspace. We calculate the discriminant d, describing the discriminating power of a given dimension, between c people in test set A.

$$d = \frac{\sum_{i=1}^{C}(m_i - m)^2}{\sum_{i=1}^{C}\frac{1}{|\Phi_i|}\sum_{x\in\Phi_i}(m_i - m)^2}, \qquad \text{where } m_i = \frac{1}{|\Phi_i|}\sum_{x\in\Phi_i} x,$$
$$m = \frac{1}{\sum_{i=1}^{C}|\Phi_i|}\sum_{i=1}^{C}\sum_{x\in\Phi_i} x \tag{10}$$

where Φ_i is the set of all class i face-key vector elements in dimension n, and m and m_i are the mean and class mean of nth dimension elements in test set A. Applying Eq. (10) to all of our surface feature maps, we see a wide range of discriminant values across the individual surface feature subspace dimensions.

Fig. 12 shows the discriminant values calculated using Eq. (10) for individual dimensions of all surface spaces produced. A dimension with a higher discriminant value has a greater ratio of between-class to within-class variance and hence termed to have greater discriminating ability, when analysed as a separated entity. For clarity, we only display ten dimensions of each space, yet the range of discriminating ability is apparent. We see that some surface spaces have a fairly uniform discriminating ability across each dimension, such as the surface space produced using the vertical curvature feature map. However, the surface space produced using other feature maps, such as max curvature, have one or two dimensions that are much more discriminating than the other dimensions of the same surface space.

It is clear that although some surface feature maps do not perform well in the face recognition tests, producing high EERs, some face-key components do contain highly discriminatory information. For example, we see that the min and max curvature features contain dimensions with a higher discriminant than the horizontal gradient and curve type dimensions, yet the EERs are significantly higher. We hypothesise that the reason for these highly discriminating anomalies, in an otherwise ineffective subspace, is that a certain surface representation may be particularly

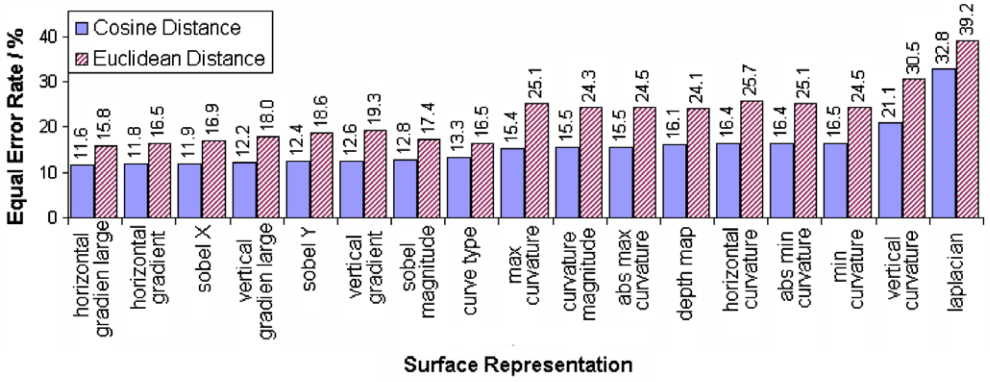

Fig. 11. Equal error rates of Fishersurface systems applied to test set A.

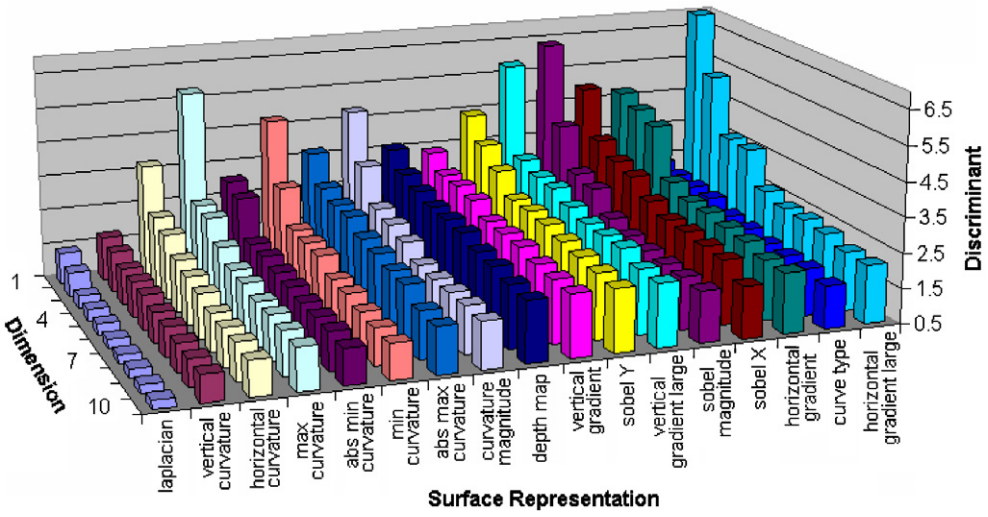

Fig. 12. Top ten discriminant values of all Fishersurface dimensions.

```
Combined surface space = first dimension of current
optimum system

Compute EER of combined surface space

For each surface representation system:

    For each dimension of surface space:

        Concatenate dimension onto combined
        surface space

        Compute EER of combined surface space

        If EER has not decreased:

            Remove dimension from combined
            surface space

Save combined surface space ready for evaluation
```

Fig. 13. Fishersurface combination algorithm.

suited to a single discriminating factor, such as nose shape or jaw structure, but is not effective when used as a more general classifier. Therefore, if we were able to isolate these few useful qualities from the more specialised subspaces, they could be used to make a positive contribution to a generally more effective surface space, reducing error rates further.

7. Combining multi-feature subspace components

In this section, we describe how the analysis methods discussed in Section 6 are used to combine multiple surface feature map subspace components into a single multi-feature face recognition system. Note that this approach first requires a subspace to be created for each specific surface feature map using LDA. One may consider this series of separate analysis procedures followed by a combination of the subspace components produced, as a sub-optimal approach, given that applying LDA directly to pre-combined (concatenated) surface feature maps would produce an optimum subspace. LDA applied to a concatenated feature space is a simpler process, however, concatenating just the 16 feature maps described in this paper would create an image of 86,400 pixels. Performing LDA on such a large image would be a process to challenge the resources of a reasonably powerful desktop computer, even by today's standards. If we were to include a greater number of surface feature maps, the problem would become intractable, although one possible approach to mitigate this would be to use data compression via principal components analysis (PCA) prior to feature map concatenation.

An alternative approach that we have found to be effective, employs an incremental process of combination, continually producing an improved system on each iteration. Once a combined system is produced, because of the progressive nature of the algorithm, another surface feature map can easily be included in the combination, without the need to restart the process. In addition, the processing time can be controlled by the size of test set A, allowing a decrease in computation time at the expense of the statistical significance of the test set.

To begin combining multiple features we must first address the problem of prioritising surface space dimensions. Because the average magnitude and deviation of face-key vectors from a range of feature maps are likely to differ by some orders of magnitude, certain dimensions will have a greater influence than others, even if the discriminating abilities are evenly matched. To compensate for this effect, we normalise moments by dividing each face-key element by its within-class standard deviation (calculated from test set A face-keys). However, in normalising these dimensions we have also removed any prioritisation, such that all surface subspace components are considered equal. Although not a problem when applied to a single surface space, when combining multiple dimensions we would ideally wish to give greater precedence to the more reliable components. Otherwise the situation is likely to arise when a large number of less discriminating dimen-

sions begin to outweigh the fewer more discriminating dimensions, diminishing their influence on the verification operation and hence increasing error rates. In Section 6, we showed how FLD could be used to measure the discriminating ability of a single dimension from any given face space. We now apply this discriminant value d_n as weighting for each surface space dimension n, prioritising those dimensions with the highest discriminating ability. This is carried out by simply multiplying each element ω_n of all face-keys $\boldsymbol{\Omega}$, by the discriminant value of the corresponding dimension d_n.

With this weighting scheme in place, we now require some criterion to decide which subspace dimensions to combine. It is not enough to rely purely on the discriminant value itself, as this only provides an indication of the discriminating ability of that dimension alone, without any indication of whether the inclusion of this dimension would benefit the existing set of dimensions. If an existing surface space already provides a certain amount of feature specific discrimination, it would be of little benefit (or could even be detrimental) if we were to introduce an additional dimension describing a feature already present within the existing set (i.e. if the information is highly correlated). The criterion required for introduction of a new dimension to an existing surface space is a resultant decrease in EER (computed using test set A), thus we have an algorithm that implements a hill climbing technique to a local optimum combination of multi-feature subspace components (Fig. 13).

We do acknowledge that other methods of dimensional combination may exist that result in superior performance to that tested here. One distinct disadvantage being that a small group of beneficial dimensions would not be included if each dimension hindered performance when introduced individually. An obvious contrast would be to begin the combination with all dimensions included, followed by an iterative elimination of dimensions. Alternatively, a genetic algorithm may result in more effective surface space combinations. However, we postpone these investigations for now, whilst we explore the simpler method in depth, before commencing with more elaborate combination methods in future work.

8. Evaluation and performance results of the multi-feature system

In order to evaluate the effectiveness of a surface space, we project and compare each facial surface with every other surface in the test set, no surface is compared with itself and each pair is compared only once. The false acceptance rate (FAR) and false rejection rate (FRR) are then calculated as the percentage of incorrect acceptances and incorrect rejections after applying a threshold. By varying the threshold, we produce a series of FAR/FRR pairs, which are plotted on a graph to produce an error curve, as seen in Figs. 15 and 16. The equal error rate (EER, the point at which FAR equals FRR) can then be taken as a single comparative value indicating the overall performance of a system, whether that be a single feature recognition system or a multi-feature recognition system.

The multi-feature subspace dimensions selected to form the combined multi-feature Fishersurface system are presented in Fig. 14. Note that different dimensions are selected depending on the distance metric (Euclidean distance or cosine distance) used to evaluate the EER.

We see that systems with lower EERs generally make the most contribution to the combined system, as would be expected. However, it is also interesting to note that even systems with particularly high EERs do contain some dimensions that make a positive contribution, although this is much more prominent for the cosine distance, show-

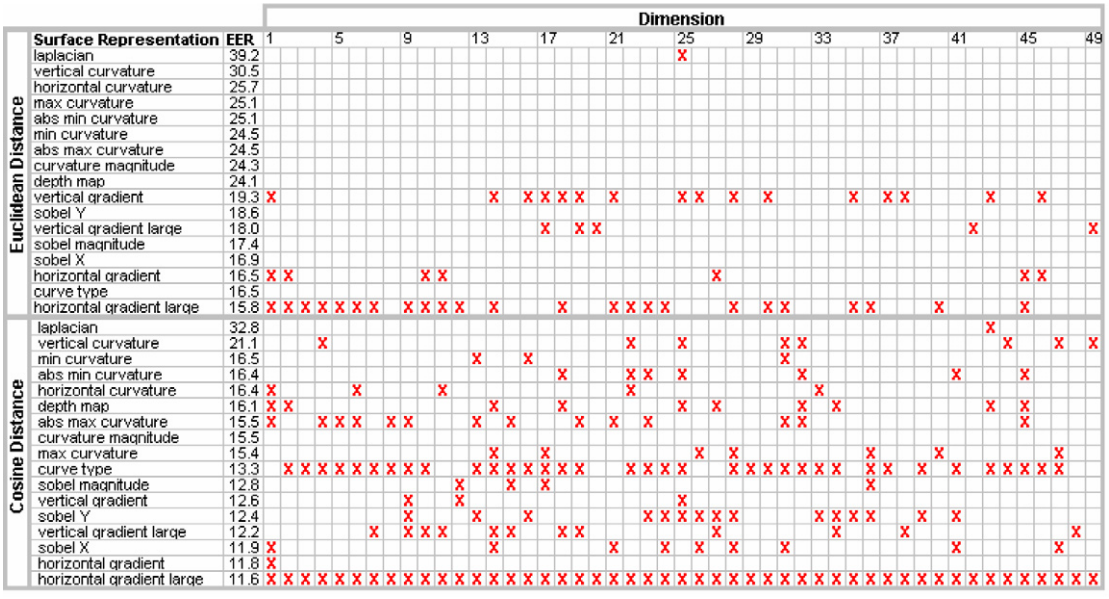

Fig. 14. Face space dimensions included (x) in the combined Fishersurface systems.

ing that this metric is more suited to combing multiple surface spaces.

Having selected and combined the range of dimensions shown in Fig. 14, we now apply these combined systems to test sets A and B using both the cosine and Euclidean distance metric. We also perform an evaluation on the union of test sets A and B: an experiment analogous to training on a database (or gallery set) of known people, which are then compared with newly acquired (unseen) images.

Figs. 15 and 16 show the error curves obtained when the best single feature Fishersurface systems and combined multi-feature systems are applied to test set A (used to construct the combination), test set B (the unseen test set) and the full test set (all surfaces from sets A and B), using the Euclidean and cosine distance metrics, respectively. We see that the combined systems produce lower error rates than the best single feature systems for all six experiments. As would be expected, the lowest error rates are achieved when tested on the surfaces used to construct the combination (7.2% and 12.8% EER, respectively). However an improvement is also seen when applied to the unseen test set B, from 11.5% and 17.3% using the best single feature systems to 9.3% and 16.3% EER for the multi-feature systems. Performing the evaluation on the larger set, providing 1,079,715 verification operations (completed in 14 min 23 s on a Pentium III 1.2 GHz processor at a rate of 1251 verifications per second), the error drops slightly to 8.2% and 14.4% EER, showing that a small improvement is introduced if some test data is available for training, as well as suggesting that the method scales well, considering the large increase in verification operations.

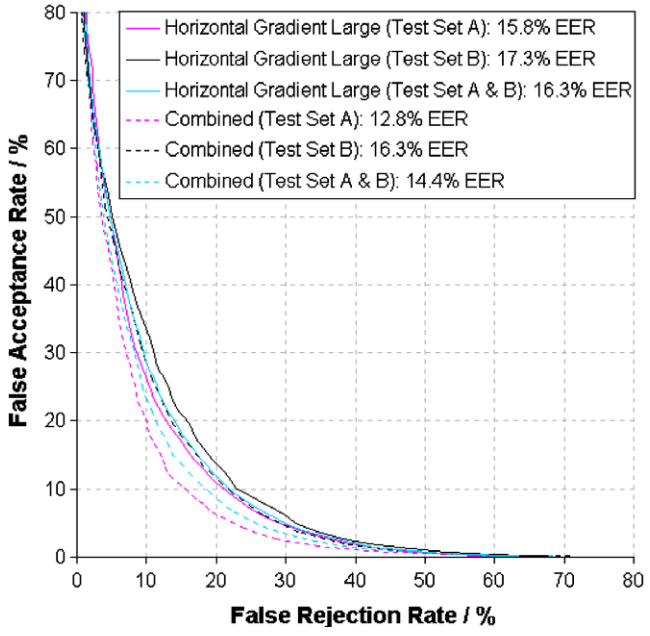

Fig. 15. Error curves comparing combined (dashed lines) and individual (solid lines) systems using the Euclidean distance metric.

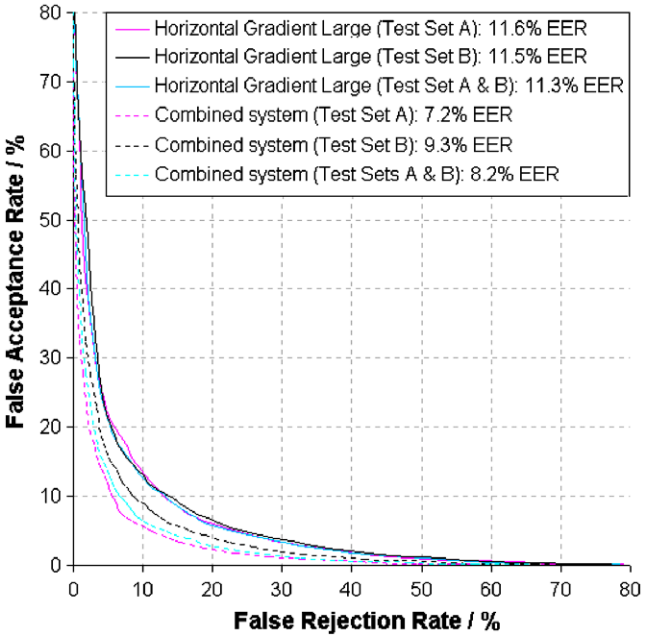

Fig. 16. Error curves comparing combined (dashed lines) and individual (solid lines) systems using the cosine distance metric.

9. Conclusion

This work is the first attempt (to our knowledge) to apply LDA to 3D facial surfaces, to apply LDA to multiple feature maps, and to combine components from multiple feature sub-spaces in order to build are more effective space in which to implement 3D face recognition.

Applying LDA to individual surface feature maps, we have shown that we can achieve reasonably low error rates, depending on the type of surface feature map used. Using FLD as an analysis tool, we have confirmed the hypothesis that although some surface feature maps may not perform well when used for recognition, they may harbour highly discriminatory subspace components that could complement other surface subspaces.

Iteratively improving error rates on a small test set, we have built up a combination of dimensions extracted from a variety of surface spaces, each utilising a different surface feature map. This method of combination has been shown to be most effective when used with the cosine distance metric, in which a selection of 184 dimensions were combined from 16 of the 17 surface spaces, reducing the EER from 11.6% to 8.2%. Applying the same combined surface space to an unseen test set of data presenting typical difficulties when performing recognition, we have demonstrated a similar reduction in error from 11.5% to 9.3% EER.

Evaluating the combined system at its fundamental level, using 1,079,715 verification operations between three-dimensional facial surfaces, demonstrates that combining multiple surface space dimensions improves effectiveness of the core recognition algorithm. Error rates have been significantly reduced to state-of-the-art levels, when evaluated on a difficult test set including variations

in expression and orientation. However, we have not attempted any further optimisation towards specific operating environments by applying additional heuristics, typically incorporated into fully functional commercial and industrial systems. The improvements made to the core algorithm will provide true benefit in a range of real-world applications, producing highly effective face recognition systems. Given the fast 3D capture method, small face-keys of 184 vector elements (allowing extremely fast comparisons), invariance to lighting conditions and facial orientation, this system is particularly suited to security and surveillance applications.

Appendix A. Extraction of surface feature maps

Depth map
The depth map is generated directly from an orientation normalised 3D face model and then used as the standard image from which all other surface feature maps are derived. This representation is highly susceptible to small translations and rotations in all directions.

Horizontal gradient
Applies the 2×1 kernel to compute the horizontal derivative describing the change in depth with respect to the x-axis. The resultant gradient map is invariant to translations along the z-axis and therefore also more stable with regard to small rotations about the x-axis. However, the small kernel size means surface noise is amplified.

-1 1

Vertical gradient
Applies the 1×2 kernel to compute the vertical derivative describing the change in depth with respect to the y-axis. Like the horizontal gradient it is invariant to translations along the z-axis, but still susceptible to noise.

Horizontal gradient large
To create this representation, we apply a similar kernel to that of the horizontal gradient representation, but calculating the change in depth over a greater horizontal distance.

-1 0 0 0 1

Vertical gradient large
Another version of the vertical gradient, using a larger 1×4.

-1
0
0
0
1

Laplacian
An isotropic measure of the second spatial derivative, calculating the depth change with respect to the x–y plane. This surface representation is invariant to translation along the z-axis and may also offer improved stability regarding small rotations about the z-axis, as it is less reliant on the vertical and horizontal direction. However, this representation is likely to significantly amplify the surface noise, creating a highly speckled texture.

0 1 0
1–4 1
0 1 0

Sobel X
Application of the horizontal Sobel derivative filter, calculating the horizontal gradient with the added benefit of local reinforcement, producing a much smoother (and potentially more robust) gradient map.

-1 0 1
-2 0 2
-1 0 1

Appendix A. (*continued*)

Sobel Y
Application of the vertical Sobel derivative filter, with similar advantages to the other gradient features, plus greatly reduced noise.

1	2	1
0	0	0
-1	-2	-1

Sobel magnitude
The magnitude of Sobel *X* and *Y* combined, creating an absolute measure of gradient magnitude with no directional bias.

Horizontal curvature
Applies the Sobel *X* kernel twice to calculate the second horizontal derivative, creating a curvature map of the 3D surface with respect to the *x*-axis. Any noise present on the surface will have been amplified by each application on the Sobel *X* filter, meaning this representation will have a high noise content.

Vertical curvature
Applies the Sobel *Y* kernel twice to calculate the second vertical derivative, creating a curvature map of the 3D surface with respect to the *y*-axis. Again, this representation will have a high noise content, due to the double amplification effect.

Curvature magnitude
The magnitude of the vertical and horizontal curvature values, providing an absolute measure of curvature magnitude with no directional bias.

Curve type
Segmentation of the surface into the eight discreet curvature types: peak, ridge, saddle ridge, minimal, pit, valley, saddle valley and flat.

Min curvature
The minimum value of the horizontal and vertical curvature maps. This representation can be thought of as a measure of surface convexity: the more convex the surface point the darker the pixel.

Max curvature
The maximum value of horizontal and vertical curvature maps. Hence, creating a representation of the surface concavity: the more concave the surface point, the brighter the image pixel.

Abs min curvature
The minimum value of the absolute horizontal and absolute vertical curvatures. The resulting representation highlights those areas that are highly curved with respect to both the *x*-axis and *y*-axis.

Abs max curvature
The maximum value of the absolute horizontal and absolute vertical curvatures. The resulting representation provides an indication of the magnitude of ridge curvature in either the horizontal or vertical directions.

References

[1] T. Heseltine, N. Pears, J. Austin, Three-dimensional face recognition: A Fishersurface approach, in: Proc. International Conference on Image Analysis and Recognition (2004).
[2] W. Zhao, R. Chellappa, 3D model enhanced face recognition, in: Proc. International Conference on Image Processing (2000).
[3] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, J. Bone, FRVT 2002: Overview and Summary, <www.frvt.org/FRVT2002/>, March (2003).
[4] S. Romdhani, V. Blanz, T. Vetter, Face identification by fitting a 3D morphable model using linear shape and texture error functions, The European Conference on Computer Vision (2002).
[5] V. Blanz, S. Romdhani, T. Vetter, Face identification across different poses and illuminations with a 3D morphable model, in: Proc. 5th IEEE Conference on Automatic Face and Gender Recognition (2002).
[6] C. Beumier, M. Acheroy, Automatic 3D face authentication, Image and Vision Computing 18 (4) (2000).
[7] C. Beumier, M. Acheroy, Automatic face verification from 3D and grey level clues, in: 11th Portuguese Conference on Pattern Recognition (2000).
[8] C. Hesher, A. Srivastava, G. Erlebacher, Principal component analysis of range images for facial recognition, in: Proc. CISST (2002).
[9] T. Heseltine, N. Pears, J. Austin, Three-dimensional face recognition: An Eigensurface approach, in: Proc. International Conference on Image Processing (2004).
[10] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: face recognition using class specific linear projection, Proceedings of the European Conference on Computer Vision (1996).
[11] A. Pentland, B. Moghaddom, T. Starner, View-based and modular eigenfaces for face recognition, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (1994).
[12] T. Heseltine, N. Pears, J. Austin, Combining multiple face recognition systems using Fisher's linear discriminant, in: Proc. SPIE Defense and Security Symposium (2004).
[13] The 3D Face Database, The University of York. Researcher Website: <www.cs.york.ac.uk/~tomh/>. Research Group Website: <www.cs.york.ac.uk/arch/>.
[14] G. Gordon, Face recognition based on depth and curvature features, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (1992).