

A 3D face matching framework for facial curves

Frank B. ter Haar*, Remco C. Veltkamp

Department of Information and Computing Sciences, Utrecht University, Padualaan 14, Utrecht, The Netherlands

ARTICLE INFO

Article history:

Received 24 July 2008

Received in revised form 20 November 2008

Accepted 22 December 2008

Available online 7 January 2009

Keywords:

Face recognition

Feature detection

Profile and contour matching

ABSTRACT

Among the many 3D face matching techniques that have been developed, are variants of 3D facial curve matching, which reduce the amount of face data to one or a few 3D curves. The face's central profile, for instance, proved to work well. However, the selection of the optimal set of 3D curves and the best way to match them has not been researched systematically. We propose a 3D face matching framework that allows profile and contour based face matching. Using this framework we evaluate profile and contour types including those described in the literature, and select subsets of facial curves for effective and efficient face matching. With a set of eight geodesic contours we achieve a mean average precision (MAP) of 0.70 and 92.5% recognition rate (RR) on the 3D face retrieval track of the Shape Retrieval Contest (SHREC'08), and a MAP of 0.96 and 97.6% RR on the University of Notre Dame (UND) test set. Face matching with these curves is time-efficient and performs better than other sets of facial curves and depth map comparison.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Before the recent developments in 3D laser scanning, the difficult task of automated face recognition was based on the comparison of 2D images. To automatically recognize a person in different images requires a system to select and match the proper set of corresponding facial features. For a 2D face recognition system to be generally applicable, it needs to cope with variances in digitizers (e.g. color, resolution, and accuracy), subjects (pose, coverage, and expression), and settings (lighting, scaling, and background). The introduction of 3D laser scanning in this area proved to be very useful, because of its invariance to setting conditions: illumination has little influence during the acquisition, the 3D measurements result in actual sized objects, and the depth information can easily separate foreground from background. 3D face information has found its application in face retrieval, face recognition, and biometrics.

1.1. Related work

The task to recognize 3D faces has been approached with many different techniques as described in surveys of Bowyer et al. [7] and Scheenstra et al. [26]. Bowyer et al. [7] divide the 3D face recognition challenge into 3D face recognition and multi-modal 3D + 2D face recognition. Although face recognition may benefit from the 2D texture information, our work can be categorized as a 3D shape based method.

An algorithm that is often applied in the context of 3D face recognition, is the Iterative Closest Point (ICP) [5] algorithm. This algorithm is able to evaluate merely the overlapping parts of two surfaces, which makes it robust to scanning deficiencies, such as missing data and outliers. However, in case of a facial expression, the acquired 3D face surface suffers from non-rigid deformations. This makes the direct application of the ICP algorithm for face recognition less reliable as shown in [3]. Mian et al. [19] proposed to extract expression insensitive regions of the face, and uses a variant of the ICP algorithm for the matching. Cook et al. [11] apply the ICP algorithm to align two 3D faces and analyze the distribution of closest point distances using Gaussian mixture models to decide if the

* Corresponding author.

E-mail address: phaar@cs.uu.nl (F.B. ter Haar).

two faces belong to the same subject. Chang et al. [10] extract multiple (overlapping) regions around the nose and match these surfaces using ICP to find similar faces. Recently, Faltemier et al. [13] developed a method that normalizes the pose of each face, extracts a predefined set of 38 surface patches, and matches pairs of patches among two scans using the ICP algorithm. To deal with facial expressions, Lu et al. [17] apply expression models to deform a neutral scan such that it fits an expression scan. For the fitting and matching they use the ICP algorithm. The main problem of the above ICP based methods, is that the time costly ICP algorithm is applied during the actual matching of 3D faces, which makes them impractical for face recognition in large sets of 3D face scans.

To improve on the face matching time, many researchers have focussed on the extraction of salient features that can be matched time-efficiently. Al-Osaimi et al. [1] developed a method that combines local and global geometric information of the face in 33 2D field histograms, applies principal component analysis (PCA) to each histogram for data reduction, and constructs a single feature vector per face scan. These feature vectors were used to recognize faces without expressions, which is still an active field of research. In methods of Blanz et al. [6] and Amberg et al. [2], a 3D morphable face model is fitted to the scan data. In the fitting process, the model's coefficients are adjusted such that the face model morphs towards the scan data. In the end, the coefficients are used in a feature vector for face recognition.

In this paper, we will focus on 3D face recognition with the use of 3D curves extracted from the 3D geometry of the face. Several *facial curve based* methods have been proposed in the past. The common goal of these methods is to extract set of 3D curves that can be effectively used for 3D face recognition. In the work of Li et al. [16], the authors extract the central profile curve and a depth contour curve from 2D depth images. They show that the recognition rate for the combination of the two curves is higher than for each curve individually. Gökberk et al. [14] published recognition results based on sets of seven vertical profile curves. Samir et al. [24] proposed a face matching method that extracts several depth contours. To match two faces, they compute the minimum energy to bend the extracted depth contours of one face to their corresponding depth contours of the other face. In [25], the same authors propose a framework that describes face surfaces with the use of geodesic contour curves and impose a Riemannian structure that measures the required energy to bend one surface to the other. They use their method to optimally deform one surface to the other, and intent to use it for face recognition purposes. Similarly, Bronstein et al. [8] compute surface geodesics for two faces, deform the surfaces such that the geodesic distances become Euclidean ones, and compare the new surfaces using a moments-based distance measure. Instead of geodesic contour curves, Berretti et al. [4] use geodesic stripes and their spatial relationship to identify faces.

All of the above mentioned curve based methods require one or more reference points, such as the tip of the nose, to start the extraction of facial curves from. Furthermore, these methods require a normalized pose of the face

scans, except for those methods that use pose invariant surface geodesics. However, to accurately locate such reference points, information on the face's pose is either used or extracted in the process. Several methods to locate the tip of the nose and to normalize the face's pose have been proposed in the literature. Li et al. [16] assume that the tip of the nose is the point closest to the scanning device. They assume an upward pose and facial symmetry to further optimize the face's pose. Mian et al. [19], assume an upward pose of the scanned subject and determine for a set of horizontal slices, what the most protrusive point is for the intersection curve. The most protrusive point among the tried slices is selected as the tip of the nose and facial symmetry is employed to normalize the face's pose. Gökberk et al. [14], use ICP to align a face scan to a face template. The assigned landmarks on the template are transferred to the scan data and facial symmetry is used to further normalize the pose and improve on the landmark locations. Chang et al. [10], subsample frontal depth images to a point where surface curvature can be effectively used for nose tip localization. Pit, peak, and saddle regions that correspond to eye corners, nose tip, and nose bridge are used to normalize the pose of the face. Such curvature information is also applied in [4,8].

The assumption of the tip of the nose being the vertex with the highest z-value is rather exceptional, because different poses, hair, clothes, noise and even expressions may interfere. Assuming an upward pose of the face in which the nose tip is the most protrusive point, does not hold in case of full head scans in which an ear or hair could easily be the most protrusive point. When the ICP algorithm is used to align a scan to a face template in order to normalize the pose, an initial alignment is required. One could use the center of mass and the principal axes of data variance, for the initial alignment, but this only works when the scan is highly similar to the template. The robustness of surface curvature heavily depends on the data density and the level of noise. The way Chang et al. [10] subsample the data density applies to depth images only.

Xu et al. [34] also pointed out some of these problems, and proposed a bottom-up approach to select the tip of the nose in a robust manner. As a first step, they select vertices among the scan that are most protrusive within a small sphere (radius 20 mm). Secondly, they use a Support Vector Machine (SVM) to select vertices that locally resemble a nose tip. Finally, they select the location with most nose tip candidates in its neighborhood as the tip of the nose. Starting from the selected nose tip, they track the nose ridge to normalize the face's pose. This method carefully discards potential nose tip locations based on local surface properties, but it fails in some simple cases where a more global notion of a face could easily improve the results. In this work we propose a method that locates the tip of the nose and normalizes the face's pose at the same time using local nose tip properties (as in [34]) in combination with a more global shape template (as in [14]).

To compare face recognition techniques, Face Recognition Grand Challenge sets [22] are publicly available. A more general Shape Retrieval Contest (SHREC) [33] has been organized in the past years to evaluate the effectiveness of 3D-shape retrieval algorithms. This contest consid-

ers 3D faces as a subset of the generic graphical models track, but also as a separate 3D face retrieval track. In this paper, we analyze 3D face models for the interrelated tasks of both face retrieval and face recognition.

1.2. Contribution

Our contributions to 3D face matching are the following. First, we introduce a new face pose normalization method that is applicable to face, full head and even full body scans in any given orientation. This method overcomes the limitations of previous methods, and it only requires a triangulated point cloud containing the tip of the nose. Second, we propose a 3D face matching framework to extract and match 3D face curves and optimize its settings. Thirdly, we evaluate sets of profiles and contours including those described in the literature. Fourthly, we propose new combinations of curves to perform both effective and time-efficient face retrieval. One of these combinations with only eight curves of 90 face samples each, achieved a mean average precision (MAP) of 0.70 and 92.5% recognition rate (RR) on the SHREC'08 face set [33] and a MAP of 0.96 and 97.6% RR on the UND face set [9].

Our face pose normalization (Section 3.1) fits 3D templates to the scan data and uses the inverse transformation of the optimal fit to normalize the face's pose. The tip of the nose is extracted from the scan data in the process. Our 3D face matching framework (Section 4) uses the nose tip as its origin and extracts a set of profile curves over the face surface. Then, it extracts samples along the profiles, which are used to determine the similarity of faces. In Section 4.4, we combine such samples in profile and contour features and select sets of features for effective and efficient face matching.

This work is an extension of the face matching framework we proposed in [29]. In our previous work, we trained and tested our feature sets on realistic synthetic data. Now, we train on this synthetic data (Sections 5 and 6), and do the testing on real face scans (Section 7). Feature sets that perform well on the synthetic face models are used for face retrieval in two popular datasets of real face scans. To cope with scanner noise and missing data, we apply an automatic mesh improvement algorithm. Furthermore, we experimented with two different distance measures, and a percentage of best matching curve samples for expression invariance. In Section 8, we compare selected feature sets with an ICP and a depth map method, and with results described in the literature.

2. Datasets

In this work we compare 3D faces from several datasets. The training sets were constructed using a morphable face model. The commonly used test sets are collections of face scans that were acquired using laser range scanners.

2.1. Training set A

The 3D faces of training set A were generated with a morphable face model, which is a point distribution model

(PDM) built from 100 face scans of the USF Human ID 3D Database [31]. To create this model an optic flow algorithm was employed to establish $n = 75,972$ correspondences among the 100 scans. Each face shape S_i was described using the set of correspondences $S = (x_1, y_1, z_1, \dots, x_n, y_n, z_n)^T \in \mathbb{R}^{3n}$ and a mean face \bar{S} was determined. Principal component analysis (PCA) was applied to these 100 sets S_i to obtain m eigenvectors of the PDM [30]. Because there are only 100 faces in the n dimensional face space, there are at most $m = 99$ meaningful eigenvectors. The mean face \bar{S} , the eigenvectors $s_i = (\Delta x_1, \Delta y_1, \Delta z_1, \dots, \Delta x_n, \Delta y_n, \Delta z_n)^T$, the eigenvalues λ_i ($\sigma_i^2 = \lambda_i$) and weights w_i are used to model new faces according to $S_{new} = \bar{S} + \sum_{i=1}^m w_i \sigma_i s_i$. In this paper, we create random instances of the morphable model by assigning m random weights w_i within the range $[-1.5, 1.5]$. Since the connectivity of the n correspondences in the PDM is known, each instance is in fact a triangulated surface mesh with proper topology and without holes.

To construct training set A, we selected seven instances of the morphable model as a query (q). Each of the queries was morphed to two other instances (i_1 and i_2) of the morphable model to create new relevant faces (r). Five intensity levels of morphing were applied, namely a 90–10, 80–20, 70–30, 60–40, 50–50 weighting scheme for the m corresponding weights (e.g. $w_i(r) = 0.6w_i(q) + 0.4w_i(i_1)$). So, for each query we have eleven relevant models including the query. The final training set consists of seven queries and 176 face instances, that is, 77 relevant models and 99 random instances. A new random pose was assigned to each instance.

Training set A is used to investigate the properties of single curves based on four sampling strategies, to lower the number of profile and contour samples for effective and efficient face retrieval, and the selection of features sets similar to those described in the literature.

2.2. Training set B

The 3D faces of training set B were generated with the same morphable face model. At first, one thousand face instances were created of which 64 were selected as a query q , 64 as instance i_1 , and another 64 as i_2 . To construct four highly relevant faces (r_h) each query was morphed in four directions with 60–40 weighting and four marginally relevant faces (r_m) using 40–60 weighing. Each query face was morphed towards i_1 and i_2 , to the mean face \bar{S} and away from \bar{S} . These additional 512 ($64 \cdot (r_h + r_m)$) face models were added to the dataset, resulting in 1512 unique 3D face models and a query set of 64 faces. A new random pose was assigned to each instance, to introduce a non-trivial pose normalization problem. This dataset was also used in "SHREC'07 – Shape Retrieval Contest of 3D Face Models" [32]. The larger embedding and the morphing towards and away from the mean face are the major differences with training set A.

Training set B is used to reassess the features sets that were selected using training set A, and to select specific sets of features that perform well. With these feature sets we perform face retrieval in our two test sets.

2.3. Test set C

As test set C, we have used the database from “SHREC’08 – Shape Retrieval Contest of 3D Face Scans” [33], which is a subset of the GavabDB [20]. This set differs from the training sets by having real laser range scans that suffer from noise and holes, and that each subject was scanned for different poses and expressions. The SHREC’08 set consists of Minolta Vi-700 laser range scans from 61 different subjects. The subjects, of which 45 are male and 16 are female, are all Caucasian. Each subject was scanned for different poses and expressions. The neutral scans include two different frontal scans, one scan while looking up ($\approx +35^\circ$), and one scan while looking down ($\approx -35^\circ$). The expression scans include one with a smile, one with a pronounced laugh, and an “arbitrary expression” freely chosen by the subject.

2.4. Test set D

The fourth dataset we use is the University of Notre Dame (UND) Biometrics Database [9]. This set consists of 953 2D depth images and corresponding 2D color textures from 277 different subjects acquired using the Minolta Vi-900 laser range scanner. All except ten scans were used in the Face Recognition Grand Challenge (FRGC v. 1). In general, the set contains frontal scans of the face with a neutral expression. However, as shown in [18] the scans show small variations in pose and expression and capture both the face and shoulders areas. To obtain 3D triangulated surface meshes, the 2D depth images were projected to 3D with the adjacent depth samples connected with triangles. The color information was neglected.

3. Preprocessing

Like many face matching algorithms, our feature extraction algorithm is pose sensitive and requires the tip of the nose as a reference point. What we need to do is to normalize the pose of each 3D face and to detect the tip of the nose. The face scans in the different datasets vary in reso-

lution and accuracy, pose and expression, and coverage of the face. We developed a method that normalizes the face’s pose in a robust manner and a mesh improvement algorithm to improve on the mesh deficiencies in face data.

3.1. Face pose normalization

Pose normalization is equivalent to correcting the viewing coordinate system that requires a view reference point, a view plane normal, and a view up vector [27]. In 3D face templates we specify the nose tip as view reference point, the gaze direction as view plane normal, and the face’s pose as view up vector. By fitting these templates to potential nose tip locations in the scan data, we eventually obtain a new coordinate system in which the face’s pose is normalized. In theory each point on a face scan can be considered as a potential nose tip location. The normal direction at a point can be estimated using the mesh’ triangles. For the training sets we considered high (positive) curvature areas as potential nose tip locations, because the tip of the nose is generally a location with high curvature. For these sets a large number of potential placements could be excluded using a curvature threshold heuristic. However, depending on the resolution and mesh quality, the curvature information can be less reliable. This is the case for the two test sets. For these scans we randomly sampled the triangular surface mesh such that every $\approx 2.0 \text{ mm}^2$ of the surface is approximately sampled once. After the selection of potential nose tip locations, 3D template matching is applied using a nose tip template t_1 to determine which potential locations locally resemble a nose tip. To the locations where t_1 fits well, a larger template t_2 is fitted to select the actual nose tip and to normalize the pose. How this bottom-up scheme solves the unknown viewing coordinate system is described below and shown in Fig. 1.

3.1.1. First template

For each of the potential nose tip locations, we have its position p and normal direction n . The first 3D template t_1 is a nose tip template with the known view reference point

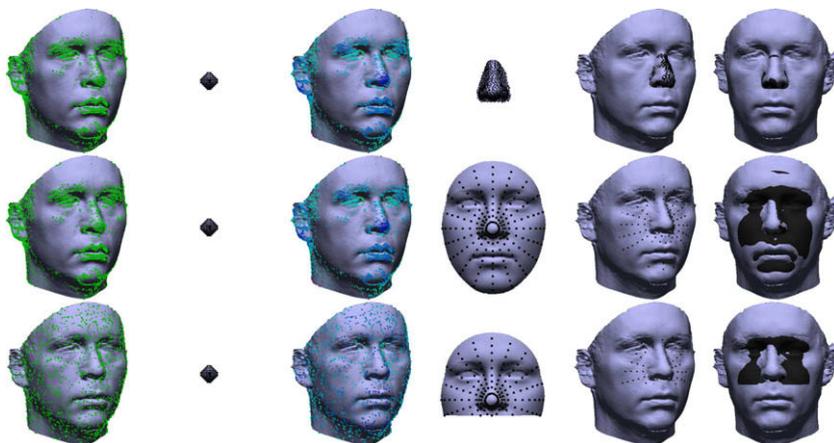


Fig. 1. Face pose normalization. From left to right, the selection of potential nose tip locations (curvature or random sampling), the small nose tip template t_1 , optimal (dark blue) locations for t_1 , template t_2 (nose or face features), best t_2 fit, and the normalized pose. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

p_{t_1} and view plane normal n_{t_1} . This template is highly symmetric around its normal n_{t_1} , which allows us to find the view plane normal while ignoring the view up vector. To fit the nose tip template to the scan data, we place the nose tip template with p_{t_1} on p and with n_{t_1} aligned to n . The alignment is refined using the Iterative Closest Point (ICP) algorithm [5], which minimizes the Root Mean Square (RMS) distance of the template's vertices to their closest points in the scan data. As a result we have for each potential nose tip a measure of how good t_1 fits that location, but also the view reference point and view plane normal defined in t_1 .

3.1.2. Second template

We reduce the number of potential nose tip locations to only a few locations around the face, where t_1 fits well. To these locations we fit a second template t_2 that has a known view reference point p_{t_2} , view plane normal n_{t_2} and view up vector u_{t_2} . Clearly, the optimal fit of this template solves the pose normalization problem. This template is placed on the remaining locations with p_{t_2} on p_{t_1} , n_{t_2} aligned to n_{t_1} , and a limited number of different view up vectors u_{t_2} . Since the angle between u_{t_2} and n_{t_2} is known, a view up vector can be instantiated using a rotation θ_{t_2} around n_{t_2} . Because the ICP algorithm is able to correct for small rotations we experimented with a new θ_{t_2} (i.e. view up vector) every thirty degrees. Each placement of t_2 is refined using ICP and the alignment with the lowest RMS distance is selected. The inverse transformation matrix for this optimal fit is used to normalize the face's pose. The point in the scan data closest to p_{t_2} is defined as the tip of the nose and used as the new origin.

Different templates based on the mean face \bar{S} were used as t_2 . For training set A, we search for the optimal placement of a nose as template t_2 , which we refer to as *nose detection*. For training set B, we search for the optimal placement of either a nose (*nose detection*) or face template (*face detection*) to normalize the pose. This is to link our pose normalization method to the face matching method. For the test sets, we used only the upper half of the face template t_2 to make the pose normalization more robust to facial changes caused by expressions (as shown in [21]). For accurate face matching it is important to acquire

the same pose for all face scans of one individual. With our template matching approach we assume that for each individual, these templates have a unique placement around the nose irrespectively to the face's proportion. This is a reasonable assumption, since the ICP algorithm minimizes the RMS distance, which enforces the alignment of typical protrusions such as the nose.

3.2. Mesh improvement

The test sets consist of triangle meshes acquired from laser range data and suffer from noise and missing data. Because our current implementation of the feature extraction requires a triangle surface mesh without holes and a proper topology, the noise needs to be removed and the holes interpolated. Furthermore, laser acquisition may even cause spikes around the tip of the nose, which in exceptional cases result in the incorrect selection of the nose tip after our pose normalization. During the mesh improvement in this section we select a new location as the tip of the nose.

Davis et al. [12] pointed out that straightforward techniques to interpolate holes in triangle meshes using curvature information or flat triangles often fail in case of complex holes. To guarantee the interpolation of all holes, we operate on 2D depth images instead of triangle meshes (see Fig. 2 for illustrations). To do so, we converted the pose normalized triangles meshes (Fig. 2a) to 1 mm × 1 mm depth images using ray casting and then we applied the following image processing techniques. First, we crop the face in the 2D depth image ($d(x, y) = z$) using an ellipse of $x^2 + 0.6y^2 = 70^2$ mm, a sphere of $x^2 + y^2 + z^2 = 100$ mm and a minimal depth of 3 mm before the tip of the nose. To improve on the nose tip location, we select a small region around the initial nose tip and take its center of mass in 2D (Fig. 2b). Secondly, to remove noise and spikes in particular, we applied a 7 mm × 7 mm median filter and a binary erosion. Thirdly, facial symmetry left and right of the nose tip was used to remove non-face data, such as hair and noisy patches, by removing pixels that are at least 10 mm closer to the viewing point than its corresponding pixel on the other side of the face. Fourthly, components with less than 200 pixels are removed (Fig. 2c).

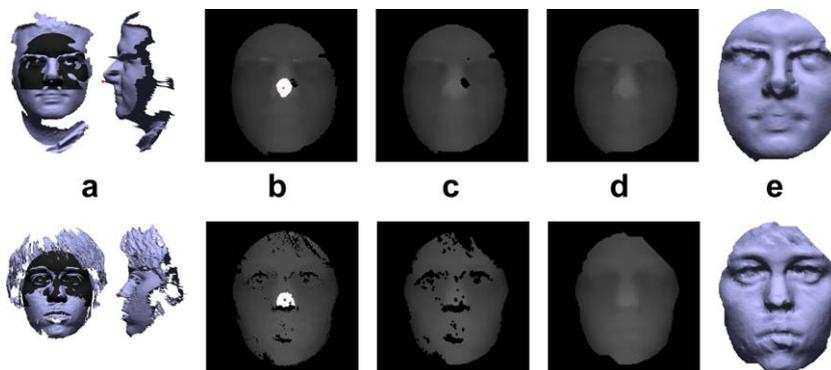


Fig. 2. Mesh improvement. Scans are pose normalized and the (incorrect) nose tips are detected (a). Each scan is converted to a depth map with a new nose tip (b), noise is removed (c), holes are filled (d), and the final depth map is converted to a new 3D mesh (e).

Finally, holes are filled effectively by linearly interpolating data around the missing data. In more detail, for each missing depth value we trace its left and right boundary and linearly interpolate these depth values in a horizontal manner. Likewise, we interpolate missing data vertically and along the two diagonals. In the end, we average the (at most) four potential depth values to fill the holes (Fig. 2d). Extrapolation of depth values is avoided, because this interpolation requires, per direction, two valid depth values of a boundary. The final 2D depth image converted to a 3D mesh with its new nose tip centered in the origin (Fig. 2e).

4. Face matching framework

Starting from the tip of the nose in a pose normalized face, our framework extracts profile curves over the face surface in different directions. These sets of profile curves are used to determine the similarity of two faces. To match two profile curves, we match a set of samples along the curves. When combined, the samples in all profiles with the same constraints build up a face contour. The facial “Z-contour” [16,24], for instance, is the curve that contains the samples from all profiles that have the same Z-value. The definition of a contour type determines which samples are extracted along the profiles (see Section 4.2).

To compute the similarity of two faces **A** and **B**, we extract N_c samples for each of the N_p profiles. Such a sample \mathbf{A}_{ij} is defined as the intersection(s) of profile i and contour j . Because the profiles and their contour samples are extracted in a structured way, we can assume that these $N_p \cdot N_c$ samples correspond for faces **A** and **B**. The distances between these corresponding samples introduce a dissimilarity. We use this information in a 3D face matching framework that consists of the generic formula

$$d(\mathbf{A}, \mathbf{B}) = \frac{1}{N} \sum_{i=1}^{N_p} \sum_{j=1}^{N_c} d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij}), \quad (1)$$

which must be instantiated with the following parameters:

- The number of profiles N_p .
- The number of contours N_c .
- The distance measure for two corresponding samples $d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij})$.

If a corresponding sample does not exist due to missing data, then we exclude that sample from the distance computation and lower the normalization factor $N \leq N_p \cdot N_c$. The complexity of our face matching framework depends on the values N_p and N_c . In case both parameters are large, then a lot of face data is used in the comparison, which is highly inefficient. With many profiles N_p and a few samples N_c , the 3D face comparison follows a contour matching approach. With a few profiles N_p and many samples N_c , the comparison follows a profile matching approach. The function $d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij})$ measures the distance between samples that correspond according to the specified contour type. The extraction and matching of feature data is described in the following paragraphs. In Section 4.4, we use our framework to evaluate different face curves and

select the most relevant profiles and contours for effective and efficient face matching.

4.1. Profile extraction

To obtain corresponding samples our framework first extracts a set of N_p profiles. A profile is defined as a 3D curve that starts from the tip of the nose and follows a path over the surface mesh with a predefined angle in the XY-plane. Such a path is defined by the intersection points of the mesh's triangles encountered along the way. Basically, we extract a profile for every $360/N_p$ degrees in the XY-plane with the tip of the nose as origin. We end a path whenever the Euclidean distance between the current location on the path and the nose tip becomes larger than 90 mm. Beyond this distance, data is less reliable because of missing data and hair. Before profile extraction, the pose normalized face is centered with its nose tip at the origin, so that the extracted sets of profiles of two different faces are aligned. We assume a proper topology of the face surface, but to be less restrictive a profile can be defined as the intersection curve of the 3D face with a plane perpendicular to the XY-plane.

4.2. Feature data

After the applied pose normalization from Section 3.1, we can assume that profiles extracted in the same direction correspond. Given two corresponding profile curves A_i and B_i , we extract N_c corresponding samples (\mathbf{A}_{ij} and \mathbf{B}_{ij}). Note that all samples are locations on the triangular surface of the face. In this work we specify four different contour samples:

- *G-samples* – samples with a shortest geodesic path of r mm over the surface to the origin.
- *C-samples* – samples with a curve distance of r mm over the profile curve to the origin.
- *XY-samples* – samples with a circular distance of $r = \sqrt{x^2 + y^2}$ mm to the origin.
- *Z-samples* – samples with a depth distance of $r = z$ to the origin.

To extract G-samples, we first computed for each vertex its geodesic distance to the origin using the fast marching method [15] and interpolated these measures for the profile paths. When a profile path is sampled using N_c G-samples (with increasing r), we refer to it as a G-profile. A G-contour is the set of N_p G-samples at the same distance r . In Fig. 3, the four different contour curves are shown.

4.3. Feature matching

The extracted $N_p \cdot N_c$ samples from one face have an assumed one-to-one correspondence to those of an other face. To match those samples we apply a symmetric distance measure that can be used for all four contour types and is rotation invariant. Therefore, we compare samples using their relative distances to the origin (i.e. tip of the nose) instead of their actual coordinates. We define the point-to-point distance (d_p) between a point p from sample

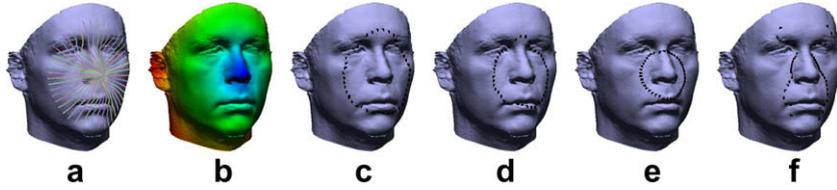


Fig. 3. From each face we extract N_p profile curves (a). A G-contour (c) is formed by selecting a sample on each profile (a) that has the same geodesic distance to the tip of the nose (colored in b). Form a G-contour (c). Other contour curves are the C-contour (d), XY-contour (e), and Z-contour (f). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

\mathbf{A}_{ij} and a point q from sample \mathbf{B}_{ij} , using the nose tip (p_{nt} = origin) and the Euclidean distance $e(p, q)$ as:

$$d_p(p, q) = (e(p, p_{nt}) - e(q, p_{nt}))^2.$$

This distance function satisfies the symmetry and non-negativity rules, but it is not a metric. Other distance measures for two 3D coordinates can be used instead. For a fair comparison of contour types, it is important that the samples are matched similarly. This is rather difficult, because a sample \mathbf{A}_{ij} can be more than one point depending on the selected contour type. The Z-contour for instance, can have multiple points p on profile A_i with a similar Z-distance to the origin. Thus a Z-sample can have multiple points, while a C-sample and a XY-sample have at most one point. To deal with multiple points per sample we define the distance d_s between two corresponding samples \mathbf{A}_{ij} and \mathbf{B}_{ij} as the smallest distance between possible point pairs

$$d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij}) = \min_{\forall p \in \mathbf{A}_{ij}, \forall q \in \mathbf{B}_{ij}} d_p(p, q).$$

In case either sample \mathbf{A}_{ij} or \mathbf{B}_{ij} is empty due to missing data, d_s is zero.

4.4. Feature selection

Our face matching framework is a useful tool to investigate the performance of profile curves and contour curves for face recognition purposes. In previous work, limited experiments were performed using either one or a few profiles and contours. With our framework we can easily select any set of profile and contour features to perform face matching with. To train and evaluate selected sets of features, we use them to query a training set. For each query, we compute its similarity to all other models in the training set, generating a ranked list of face models sorted on decreasing similarity values in the process. For each ranked list we compute the average precision, which is the average of precisions computed at those ranks where a relevant face is found. The precision (P) and average precision (AP) within the scope of retrieved items are defined as,

$$P(\text{scope}) = \frac{\sum_{s=1}^{\text{scope}} \text{rel}(s)}{\text{scope}}$$

$$AP(\text{scope}) = \frac{\sum_{s=1}^{\text{scope}} \text{rel}(s) \times P(s)}{\sum_{s=1}^{\text{scope}} \text{rel}(s)},$$

where $\text{rel}(\text{rank}) = 1$ if the face on the specified rank is relevant and zero otherwise. The scope we use, equals the size of the dataset. The average precision emphasizes the early retrieval of relevant faces, which is a desired property

for our feature based face retrieval. The mean average precision (MAP) over all queries is used to assess a selected feature set.

5. Training on set A

In this section, training set A is used to investigate the properties of single curves based on different feature data. Furthermore, we experiment with the number of extracted profiles N_p and contours N_c , to find subsets of curves for both effective and efficient face retrieval.

5.1. Single curve matching

For efficient face matching, previous work aims at reducing face information to a single distinctive curve. With our framework we can extract a single contour or profile curve, and assess its performance on our training set. We tested the robustness of single contours under varying conditions that are common in practice, such as small errors in nose tip localization and pose normalization, and different levels of noise. This was done by evaluating the MAP of each contour within the range $r = [1, 140]$ mm. Fig. 4 shows the following results for each of the contour curves:

- *Basic* – matching original query and database faces with known pose and nose tip location.
- *Tip* – the queries were disrupted with a nose tip displacement, $tip_1 = 2$ mm and $tip_2 = 4$ mm from the actual nose tip.
- *Rot* – the queries were disrupted with an Euler rotation (ρ, ρ, ρ) , rot_1 with $\rho = 1$ and rot_2 with $\rho = 2$ degrees.
- *Noise* – the queries were disrupted with additional noise relative to the average edge length η in the mesh, $noise_1$ with 0.1η and $noise_2$ with 0.2η .
- nd_o – matching original query and database faces after automatic nose detection to normalize the pose and to localize the nose tip (Section 3.1).
- nd_r – matching randomly rotated query and original database faces after automatic nose detection.

From these results we learn the following:

- (1) Small changes in r can cause a large decrease in performance.
- (2) C-contours are more robust to errors in nose tip localization and pose normalization, and XY-, and Z-contours are more robust to noise.

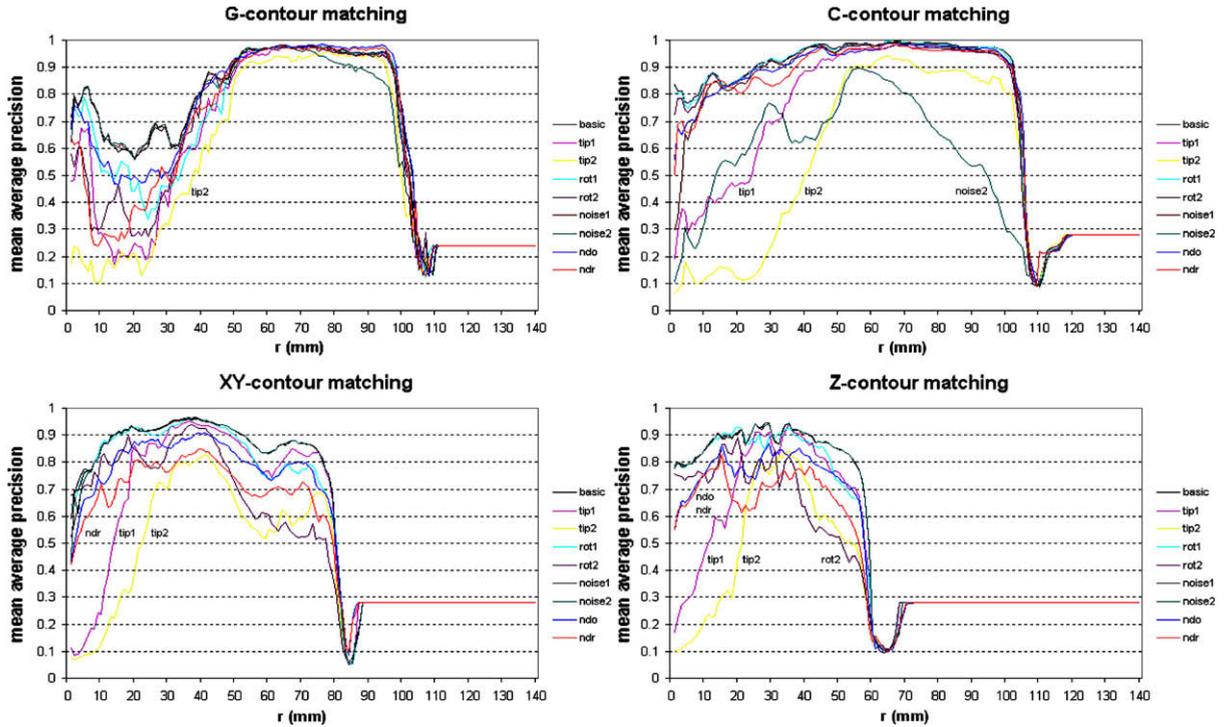


Fig. 4. Training set A. The mean average precision graphs of single G-, C-, XY-, and Z-contours for sample values 1 to 140 mm under varying conditions.

- (3) G-contours on the outer regions of the face are robust under all these conditions.
- (4) Each contour type has an *active region* of $r = 1$ mm to the r just before the MAP drops to a minimum, beyond that point a contour lacks sample data because of the face cropping.

The *basic* results can be used as a reference for the optimal results. The *ndo* results are comparable to methods that assume scans to be faced forward. For 3D or 2.5D face retrieval the *ndr* results are important, because this involves pose normalization of faces (or head models) under all possible orientations.

To investigate how well a single profile can be used for the purpose of 3D face retrieval we plot the MAP of each sampling strategy per angle within the range $a = [0, 359]$. Each profile is actively sampled with a large number of contour samples, that is 60 to 105 depending on the active region of a contour type. Profile matching suffers from disrupted queries in a similar manner as contour matching. Therefore, we show only the *basic* results and the results after automatic nose detection (*ndo*, *ndr*). Fig. 5 shows that the maximal performance for a single contour is higher than for a single profile, which means that a single contour can be more descriptive than a single profile.

5.2. Multiple curve matching

Single curve matching has regions for which curves are able to obtain high performances, but a small change in range r or angle a can cause a large decrease in perfor-

mance. In other words, effective face retrieval based on a single curve has a small chance of success. In this section, we assess face matching using multiple curves, based on the *basic*, *ndo*, and *ndr* results from querying training set A.

To achieve effective face retrieval, using data from multiple curves is essential. However, there is a trade off between the effectiveness and efficiency. Parameters N_p and N_c of our framework determine the amount of samples used to describe a face. A first step is to decrease these numbers to a point where face matching is still effective, but more efficient. To do so, we extracted $N_p = 360$ profile curves and sampled each profile with $N_c = 360$ contour samples equally spaced over the *active region* (see previous section). From these 360 profiles and contours we selected subsets with a decreasing amount of samples $N_p \cdot N_c = n_f \cdot n_f$ with $n_f = \{360, 180, 90, 45, 24, 20, 16, 12, 8, 4\}$. Note that a set of $360 \cdot 360$ surface samples exceeds the number of vertices in our face models.

From the results in Fig. 6 we learn that the number of samples can be reduced from $360 \cdot 360$ to $45 \cdot 45$ without losing discriminative power. Compared to the number of vertices of a face model, $45 \cdot 45$ samples is already a large reduction of face data. With a number of profiles $N_p = 45$ we can investigate the performance of *multiple contours* by varying $N_c = \{360, 180, 90, 45, 24, 20, 16, 12, 8, 4\}$ and the other way around for the retrieval performance using *multiple profiles*. Fig. 6 shows, in general, higher performances for the use of multiple curves compared to the use of a single curve. For a small number of curves the use of multiple contours outperforms the use of multiple profiles.

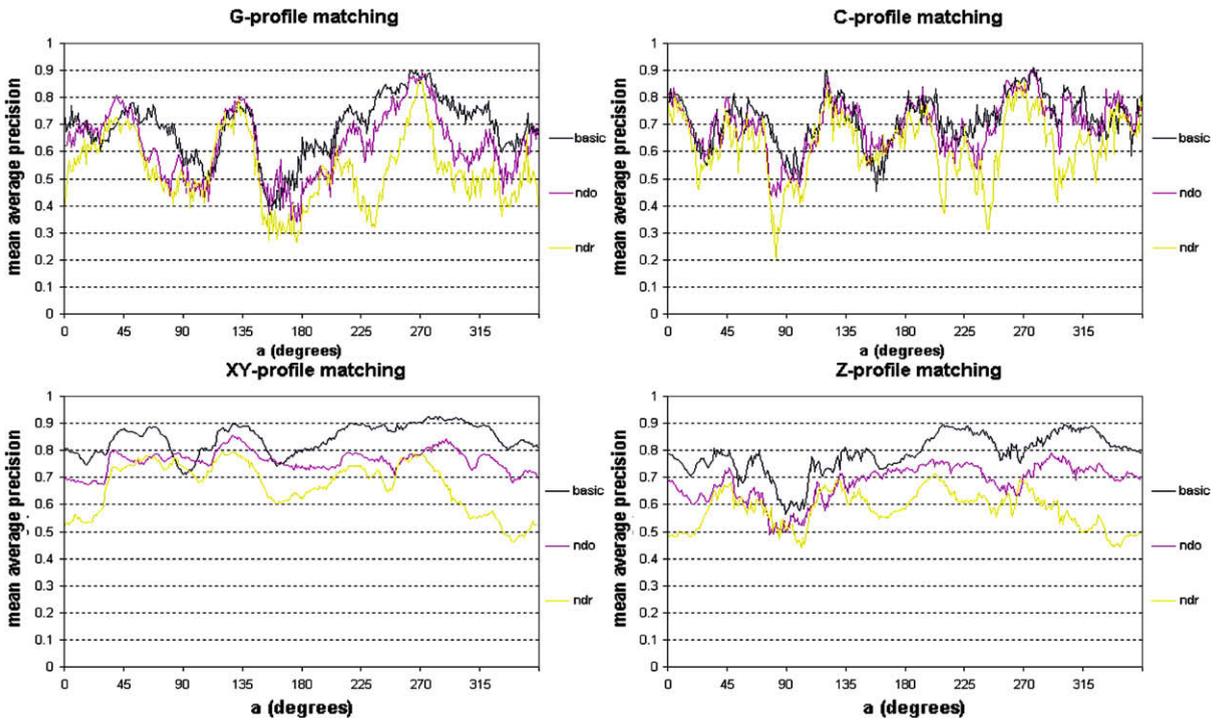


Fig. 5. Training set A. The mean average precision graphs of single G-, C-, XY-, and Z-profiles for angles 0 to 359 degrees after automatic nose detections.

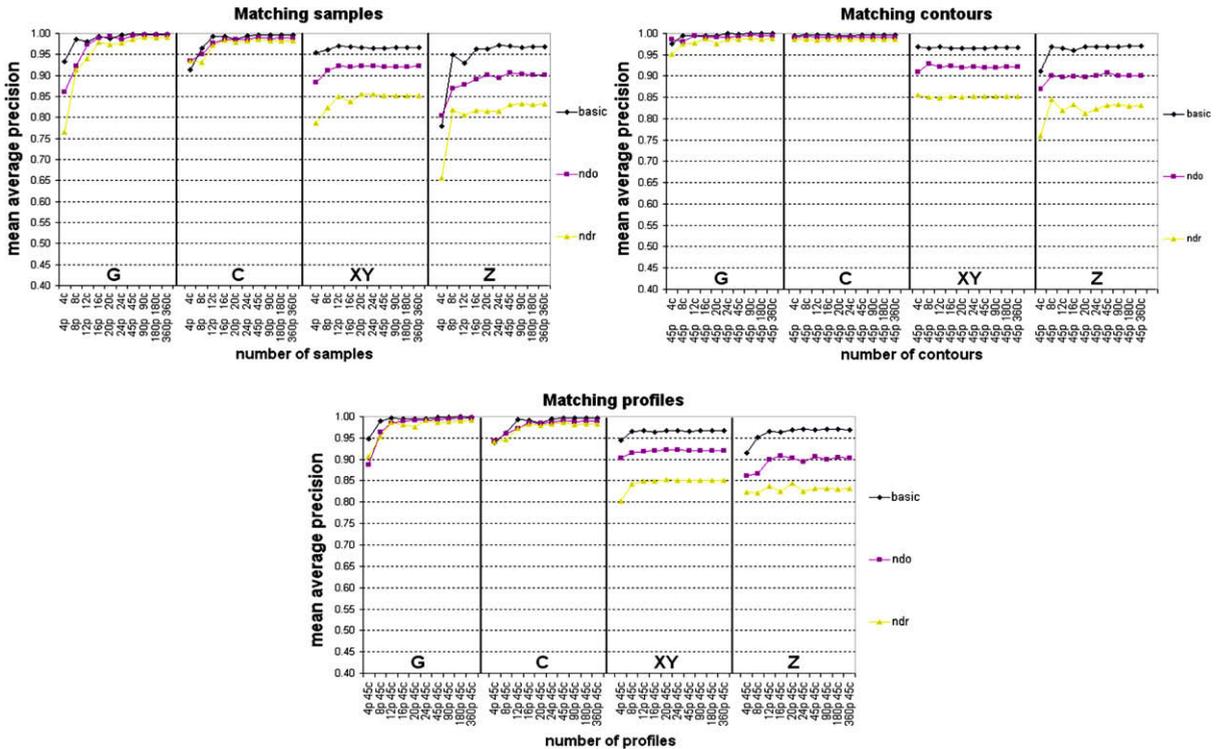


Fig. 6. The performance while varying the number of samples, contours, or profiles.

5.3. Central profile and optimal contour

In the work of Li et al. [16] face recognition is performed on 2D depth images using the combination of a single

Z-contour at distance $z = 30$ mm with the central profile curve from forehead to chin. With our framework we can perform face matching similarly by selecting the same Z-contour, the XY-profile from nose to forehead, and the

XY-profile from nose to chin. Furthermore, we can manually select the best G-, C-, XY-, and Z-contour and the central G-, C-, XY-, and Z-profiles for training set A. We can combine these *manually selected* contours and profiles among different sampling strategies, which we refer to as *hybrid matching*. We selected the following contour curves with the highest MAP for the *basic* results (see Fig. 4): the G-contour at 77 mm, the C-contour at 68 mm, the XY-contour at 36 mm, and the Z-contour at 35 mm. These contour curves are shown in Fig. 3c–f.

In this section we explore the 16 hybrid combinations of the G-, C-, XY-, and Z-contour ($N_p = 45$) and central G-, C-, XY-, and Z-profiles ($N_c = 45$). Results on training set A (Fig. 8) show a high performance for the combinations of the G-, C-, and XY-contour with G-, C-, and XY-profiles. The marked areas show common factors of the results. For the training set the G and C-curves perform best followed by XY-curves and then Z-curves. Li's combination of the two vertical XY-profiles and one Z-contour performs reasonably well, but not as good as our manually selected contours and profiles. Of course the set of three optimal curves may differ per training set.

5.4. Optimal contours

From Section 5.2, we have learned that for a small number of curves the contours are more distinctive than profiles. Instead of combining one contour with two profiles, it makes sense to combine the optimally selected contours. Fig. 8 shows the combined performance of the optimally selected contour curves. Results show that the Z-contour has a negative influence on the overall performance of selected features. Nevertheless, the *basic* results for the Z-contour are high, so its lower performance is probably caused by its lack of robustness to even small changes in the face's pose.

5.5. Eight contours

From the results in Section 5.2 we have learned that eight uniformly selected curves having 45 equally spaced samples has a reasonable performance. Thus, with only 360 samples per face we are already able to perform effective face matching. The single curve properties from Section 5.1 showed that each sampling type (G, C, XY, and Z) has its strengths and weaknesses. So, it makes sense to investigate the performance of *hybrid matching* using two profiles and one contour based on different sampling types as we did in Section 5.3. In the following experiment we have used hybrid combinations of eight contour curves in an attempt to improve the performance.

With the use of our framework we generated for each contour type, four equally spaced contours with $N_p = 45$ samples (see Fig. 7). These G-, C-, XY-, and Z-contours were then combined into ten unique feature sets. Because the combination of four G-contours with the same four G-contours (and the three other exact matches) is useless, we use eight equally spaced contours instead. Fig. 8 shows the results of the ten unique combinations of G-, C-, XY-, and Z-contours. The results from this experiment show a high performance for sets of eight G-contours, eight C-contours,

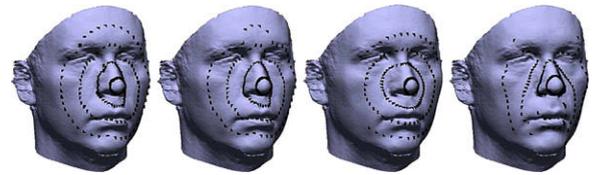


Fig. 7. Features used in hybrid matching. From left to right, four G-, C-, XY-, and Z-contours.

and the combination of four G-contours and four C-contours. Reasonable results are obtained for combinations of XY-contours with either G- or C-contours.

6. Training on set B

In the previous sections we used training set A to investigate the performance of single curves under varying conditions. Training set A was also used to show the effect of lowering the number of profiles N_p and contours N_c , such that face matching became more efficient with the same high performance. In this section we evaluate the selected feature sets from Sections 5.3–5.5 on training set B, which has a larger embedding and a larger variety of relevant faces per query. For each set of features we show its *basic* and nd_r results in Fig. 8. To obtain the *basic* results, we used the ground truth information of this dataset to undo the applied rotations and to select the nose tip locations. These results indicate the optimal performance that can be reached when the correct pose and nose tip are found. The nd_r results were obtained by applying nose detection to faces of the dataset directly. Observations that count for all *basic* and nd_r results are: (1) The larger embedding and the greater dissimilarity of relevant faces decreases the overall mean average precision from around 0.9 for training set A to around 0.7 for training set B. (2) The performance gap between results after the applied nose detection (nd_r) and the predefined pose and nose tip (*basic*) shows that our 3D face retrieval can be further improved with optimized pose normalization and nose tip localization.

To confirm the latter observation we performed an additional experiment using the face template instead of the nose template and more view up vectors (every ten degrees) as described in Section 3.1. To evaluate the pose normalization and nose tip localization results, we used the ground truth data to determine the difference of the face's pose and the located nose tip. For local nose detection nd_r the mean and standard deviation of this evaluation are respectively 3.0 ± 1.9 degrees, and 1.4 ± 0.95 mm. For global face detection fd_r these results are respectively 2.1 ± 1.4 degrees, and 1.2 ± 0.83 mm. Global face detection results in a better face pose normalization and a more accurate nose tip localization, which has a positive effect on the retrieval performance.

6.1. Central profile and optimal contour

The combination of the optimally selected G-, C-, and XY-contour with the central XY-profile performs well on

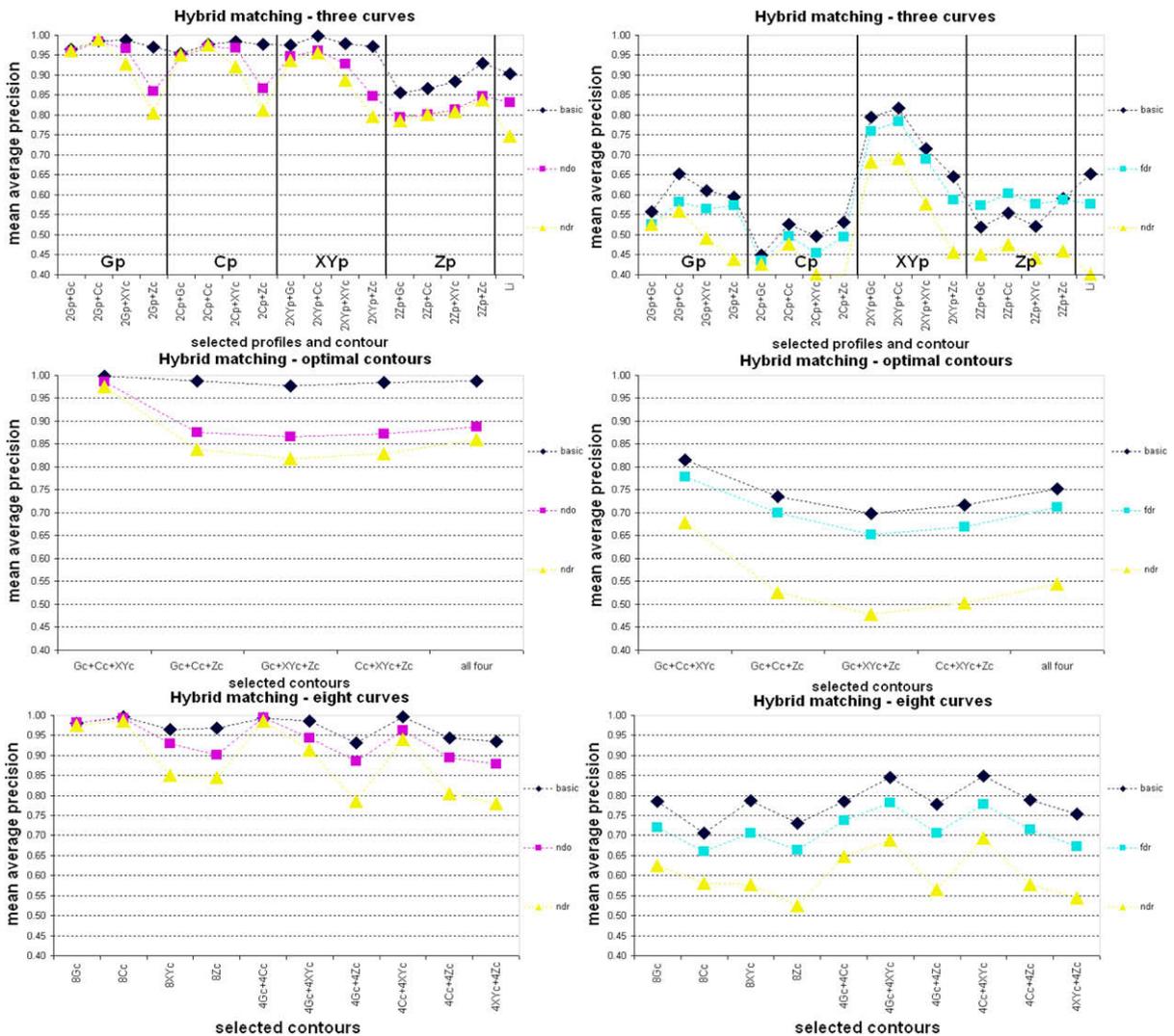


Fig. 8. Retrieval results on training set A (left) and training set B (right). The framework was applied using combinations of manually selected curves (top), middle), and eight uniformly selected curves (bottom).

training set B, as it did on training set A. Remarkable is that the G- and C-profiles show a relatively large decrease in performance compared to the results from training set A. The C-profile performed very well on that training set because all models had a similar level of noise. Training set B on the other hand, contains relevantly classified faces which were morphed towards and away from the mean face introducing different levels of noise. This and the fact that C-profiles are less robust to noise explains this drop in performance. The decrease of the G-profile’s performance can be explained as follows. G-contours close to the tip of the nose, range $r = 0$ mm to $r = 40$ mm, are not effective to retrieve relevant faces (see Fig. 4). However, a G-profile contains samples within this range, which makes it less reliable. For the rather small training set A the central G-profiles were discriminative enough, but the larger embedding of training set B caused a lower performance of these curves. The XY-profile, with its constant performance and high robustness to noise, is therefore the best type of pro-

file curve. The combination of the two central XY-profiles and optimal C-contour obtained a MAP of 0.69 for nd_r and even 0.78 for fd_r .

6.2. Optimal contours

The results for optimally selected contours on the training set B are similar to those on training set A. Again the highest results are obtained for the combined G-, C-, and XY-contour. The MAP in this case are 0.68 for nd_r and 0.78 for fd_r .

6.3. Eight contours

For feature sets of eight uniformly selected contour curves, results shows again a drop in performance for the combinations of C-contours and G-contours. For the eight G-contours, the ones closest to the nose tip have a negative influence on the performance. For the eight C-contours, the

noise is again the cause. Nevertheless, the highest results ($fd_r = 0.78$) are obtained for the hybrid combinations of four XY-contours with either four G-contours or four C-contours. Since C-contours and G-contours are very much alike, their combination does not improve the performance. The most important observation is that for each set of eight single type contours, there is a hybrid combination of eight contours with a higher performance. This means that hybrid combinations can improve on the effectiveness of face retrieval, without losing efficiency.

Based on the results of training set B, we selected several feature sets to test on our two test sets with real face scans. For their high efficiency, we selected the combinations of the central XY-profile and our optimal C-contour (2XYp + Cc), the central XY-profile and Li's optimal Z-contour (Li), and our optimal G-, C-, and XY-contour (Gc + Cc + XYc). For their high performance, we selected hybrid combinations of four XY-contours with either four G-contours (4Gc + 4XYc) or four C-contours (4Cc + 4XYc). For comparison, we selected the sets of eight uniformly selected G-contours (8Gc), G-contours (8Cc), G-contours (8XYc), and Z-contours (8Zc). These results on training set B are also listed in Table 1. The *basic* results shown in this table indicate that the retrieval performance can be further increased, when an even more accurate pose normalization and nose tip localization method is applied.

7. Test results

In the previous section, we selected features with a high retrieval performance in a training set with a large embedding. In this section, these selected features are used for the retrieval of relevant faces in test set C and D. The scans in these test sets were all pose normalized and cleaned as described in Section 3. Test set C contains for each of the 61 subjects seven different scans. In this data set of 427 scans, each scan is used as a query once. Test set D contains 953 scans, which are all used to query set D. The MAP over all queries is used as a measure to evaluate the retrieval of relevant faces in a certain test set. Relevant faces are scans acquired from the same subject as the query scan.

So far, we have used only a single distance measure d_p to compute the distance between corresponding samples. Namely, the squared relative Euclidean distance to the tip of the nose d_{p1} ,

$$d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij}) = \min_{\forall p \in \mathbf{A}_{ij}, \forall q \in \mathbf{B}_{ij}} d_p(p, q),$$

$$d_{p1}(p, q) = (e(p, p_{nt}) - e(q, p_{nt}))^2.$$

Table 1

Retrieval results of selected features on training set B.

Features	Samples	nd_r	fd_r	<i>Basic</i>
2XYp + Zc(Li)	135	0.40	0.58	0.65
2XYp + Cc	135	0.69	0.78	0.82
Gc + Cc + XYc	135	0.68	0.78	0.82
8Gc	360	0.63	0.72	0.79
8Cc	360	0.58	0.66	0.71
8XYc	360	0.58	0.71	0.79
8Zc	360	0.52	0.66	0.73
4Gc + 4XYc	360	0.69	0.78	0.85
4Cc + 4XYc	360	0.69	0.78	0.85

Now, we also use the more commonly used squared Euclidean distance d_{p2} ,

$$d_{p2}(p, q) = (e(p, q))^2.$$

Table 2 shows that especially the eight uniformly selected G- and C-contour benefit from distance measure d_{p2} . The performance slightly decreases for our optimally selected contours, which were selected based on the initial distance measure d_{p1} . This is clearly a disadvantage of manually selecting curves.

The retrieval results on test set C are not very high. Up till now, we used our framework to select subsets of contours that can be used for effective and efficient face matching under facial morphing. However, the triangle meshes in test set C are distorted by changes in facial expression. For expression invariant face retrieval, our framework needs to cope with these changes. Samir et al. [24] specifically selected a subset of Z-contours that are reasonably robust for a dataset of six different expressions per person. Mian et al. [19] applied a variant of the ICP algorithm to match only a masked face region (nose, eyes and forehead) which is assumed to be static under facial expressions. Instead of restricting our framework to a specified subset of facial curves, we can select a percentage of profiles that matches best for two input faces. This way, a person might even be recognized using profiles that go through the mouth, in case that region remains unchanged from one expression to another. To do so, we introduce function f_w in our generic formula (Eq. (1)), so that a weight can be assigned to specific profile curves:

$$d(\mathbf{A}, \mathbf{B}) = \frac{1}{N} \sum_{i=1}^{N_p} f_w \sum_{j=1}^{N_c} d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij}). \quad (2)$$

Here, we simply use f_w to assign a weight of 1 to a percentage of best matching profiles and a weight of 0 to the other profiles. Percentages of 25%, 50%, 75%, and 100% were used. With this function, profiles in facial areas that changed because of an expression can be neglected for the actual face comparison. We intentionally use a percentage of profiles and not a percentage of samples, because samples close to the nose have a smaller distance than those far from the nose but are often less distinctive for face retrieval. As a result, it is not possible to use this face matching scheme for single profile curves such as the central profile (2XYp + Cc and 2XYp + Zc(Li)). Matching two single con-

Table 2

The MAP results on test sets C and D using two different distance measures.

Features	Test set C		Test set D	
	d_{p1}	d_{p2}	d_{p1}	d_{p2}
2XYp + Zc(Li)	0.52	0.55	0.83	0.87
2XYp + Cc	0.59	0.58	0.93	0.93
Gc + Cc + XYc	0.60	0.57	0.94	0.94
8Gc	0.59	0.64	0.94	0.96
8Cc	0.57	0.65	0.89	0.95
8XYc	0.65	0.65	0.95	0.95
8Zc	0.56	0.57	0.89	0.91
4Gc + 4XYc	0.66	0.66	0.94	0.94
4Cc + 4XYc	0.66	0.66	0.94	0.94

tours of faces **A** and **B** using 50% of its best matching ‘profiles’, means that only half of the contour’s corresponding samples are used to compute the similarity. The number of profiles N_p is increased to $N_p = 90$, to compensate for the reduction of matched data. The retrieval results based on measure d_{p2} and various percentages are shown in Table 3.

Results show that the use of a smaller percentage of best matching profiles is an easy way to improve on expression invariant face retrieval. For test set C, the average performance of contour features improves from 0.64 at 100% to 0.67 at 50%. For test set D that has fewer expression scans, the effect is less significant with 0.94 at 100% and 0.95 at 50%. Fig. 9 shows an example of expression invariant face retrieval.

8. Method comparison

The results achieved on test set C can be compared with the results from SHREC’08 [33]. Test set D (UND) is used under various conditions in previous work. For a direct comparison of our facial curves method, we implemented two comparable methods.

After the mesh improvement of Section 3.2, a cleaned depth map is acquired, which is also represented by a triangle mesh (see Fig. 2). The first method directly compares the distances of the depth maps (after aligning the nose tips) and computes the RMS distance for a percentage of the best matching samples. Only foreground pixel-pairs are considered. Note that this *depth map matching* is very similar to the comparison of a large set of XY-features. The second method is an *ICP matching* method. This method uses in each iteration a percentage of the closest point pairs to minimize the RMS distance between two triangular meshes, refining their alignment in the process. To have a symmetric distance measure, closest point pairs (p, p') are selected for face **A** to **B** and vice versa. Closest point pairs for which p' belongs to the surface boundary of a face scan are not used in the distance measure. To have a time-efficient measure, the RMS distance after just four iterations was used to match two faces, only vertex-vertex pairs were considered, and kD-trees were used for fast closest point operations. These are all common variations of the ICP algorithm [23]. Note that, the accurate pose normalization allows for the small number of iterations.

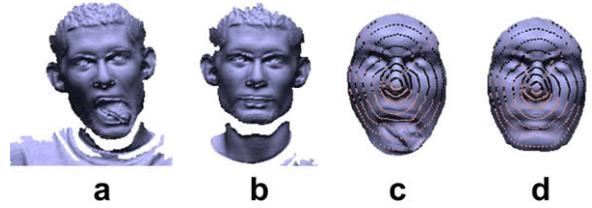


Fig. 9. Expression invariance. A face scan with expression (a) that has a relevant scan on top of the ranked list (b) for the set of eight G-contours, because only 50% of its best profiles (in black) were matched (c, d).

Despite all efforts to make the *depth map matching* and *ICP matching* time-efficient, they were too slow for practical use. Therefore, we subsampled the $1 \text{ mm} \times 1 \text{ mm}$ depth maps of Fig. 2 to $3 \text{ mm} \times 3 \text{ mm}$ depth maps, and corresponding surface meshes with approximately 1800 valid data points. For a fair comparison, our curve matching was based on these lower resolution surface meshes as well, which even turned out to result in higher MAPs. For all methods we use the squared distance between samples, which means we again use d_{p2} for our curve matching. We report the MAP results based on percentages 25%, 50%, 75%, and 100% for test sets C and D, in Table 3. Results in this table show that from all selected sets of profiles and contours, the set of eight uniformly selected G-contours (50%) performs best, closely followed by the other sets of eight contours except the eight Z-contours. This set of contours outperforms the depth map matching in both efficiency as effectiveness. ICP matching has a higher performance, but already requires 16.8 s to query a dataset of 1000 faces once.

The test sets C and D are usually used in the context of 3D face recognition. To compare the performance of selected feature set with the performance of other methods described in the literature, we determine recognition rates. The *recognition rate* (RR) is the ratio of relevant faces (other than the query face) retrieved on top of the ranked lists. When a face matching system retrieves for each query one of its relevant database models on top of the list, the RR is 100%. To increase the flexibility of a face recognition system, a relevant face within a small scope could be allowed for a successful recognition instead. The *cumulative match characteristic* (CMC) curves in Fig. 10, show the recognition rates for these larger scopes. The faster a CMC

Table 3

The MAP results on test sets C and D using contour ($N_p = 90$), ICP, and depth map matching. Timings are reported in seconds per 1000 matches.

Features	Test set C				Test set D				Samples	Time
	25%	50%	75%	100%	25%	50%	75%	100%		
2XYp + Zc(Li)				0.55				0.86	270	1.3
2XYp + Cc				0.60				0.93	270	1.3
Gc + Cc + XYc	0.60	0.62	0.61	0.60	0.94	0.95	0.95	0.95	270	1.3
8Gc	0.70	0.70	0.70	0.67	0.95	0.96	0.96	0.96	520	3.1
8Cc	0.69	0.69	0.68	0.67	0.95	0.96	0.96	0.95	520	3.1
8XYc	0.67	0.68	0.67	0.65	0.94	0.95	0.95	0.95	520	3.1
8Zc	0.67	0.64	0.61	0.57	0.92	0.94	0.94	0.90	520	3.1
4Gc + 4XYc	0.67	0.69	0.68	0.67	0.93	0.95	0.95	0.94	520	3.1
4Cc + 4XYc	0.68	0.69	0.69	0.68	0.94	0.95	0.96	0.95	520	3.1
ICP	0.75	0.78	0.73	0.71	0.96	0.97	0.98	0.97	1800	16.8
Depth map	0.49	0.53	0.56	0.61	0.80	0.84	0.87	0.89	1800	7.8

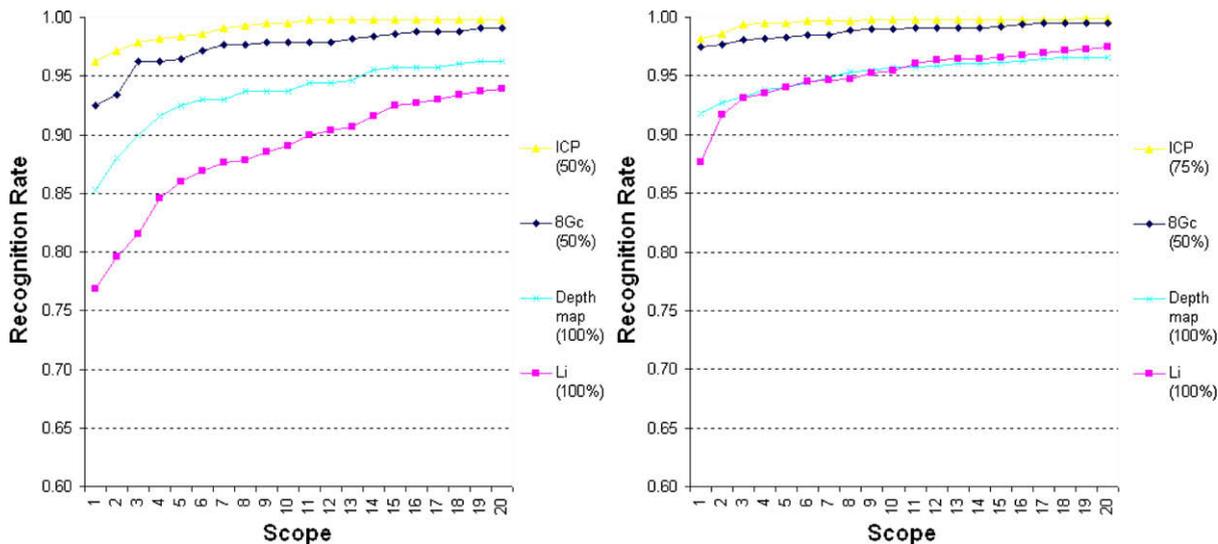


Fig. 10. The CMC curves for test sets C and D using facial curves, ICP, and depth map matching.

curve approaches one, the better that matching algorithm is.

The results we achieved on test set C, can be compared to the results reported in SHREC'08. With a MAP of 0.70 and 92.5% RR, our set of eight G-contours (50%) performs reasonably well. These results are higher than the facial curves of Li (2XYp + Zc) with a MAP of 0.55 and 74% RR, higher than our depth map method (100%) with a MAP of 0.61 and 85.2% RR, but lower than our ICP based method (50%) with a MAP of 0.78 and 96.3% RR. These differences in performance are also shown in Fig. 10 using CMC curves. In these graphs, the first point of each curve shows the reported rank-first recognition rates. Compared to the results reported in [28], we notice that Morphable model based face matching by Amberg and the use of iso-geodesic stripes by Berretti have a higher performance with MAPs varying from 0.65 up to 0.94. Moment invariants by Xu, and region based matching by Nair have a lower performance with MAPs varying from 0.46 up to 0.66.

For test set D we achieved a MAP of 0.96 and 97.6% RR using eight G-contours (50%), Li's curves 0.86 and 87.7%, our depth map method (100%) 0.89 and 91.8%, and our ICP based method (75%) 0.98 and 98.2%, respectively. The recognition rates were computed over 876 queries, because 77 subjects have only one scan in this data set. The CMC curves of these methods are also shown in Fig. 10. For comparison, Blanz et al. [6] achieved a 96% RR for 150 queries in a subset of 150 faces using 1000 morphable model coefficients. Samir et al. [24] reported a 90.4% RR for 270 queries in a subset of 470 scans using a facial depth curves. Mian et al. [18] reported a 86.4% RR for 277 queries in a subset of 277 scans using tensor matching.

9. Concluding remarks

In this work we proposed a new pose normalization method and a 3D face matching framework. Pose normalization is performed by fitting 3D templates to the scan data and using the inverse transformation of the best fit

to normalize the pose. The fitted template is used to extract the tip of the nose. Starting from the tip of the nose we extracted a set of profile curves, which were sampled using G-, C-, XY-, and Z-samples. The number of profiles, the number of contours samples, and the distance measure are the parameters to instantiate our framework. According to the selected settings, our framework extracts corresponding samples from faces and matches them using the defined distance measure.

Throughout this work we have optimized our framework's settings for efficient and effective face matching. Furthermore, we examined the properties and the face retrieval performance of single facial curves, uniform selected curves, manually selected curves, and hybrid combinations, using different types of curve samples. This evaluation was done for training set A and to a certain extent for training set B as well. Based on these results we were able to select a few feature sets with a high retrieval performance. These feature sets were used in two different test sets with real face scans, namely the SHREC'08 and UND face sets. After pose normalization and face scan improvement, we applied our feature sets for the retrieval of relevant face scans. Besides that, we also applied a depth map and an ICP based method. To cope with the expressions in the face scans, we used a percentage of face data that matched best.

The combination of the central profile with a our optimally selected G-contour performed better than the same central profile with Li's optimally selected Z-contour. With the same amount of feature points, we achieved a higher performance using an optimally selected G-, C-, and XY-contour. In the end, the eight uniformly selected G-contours using 50% of the $N_p = 90$ best matching profiles performed slightly better than other combinations of profiles and contours. With a MAP of 0.70 and 92.5% RR on test set C (SHREC'08), and a MAP of 0.96 and 97.6% RR on test set D (UND), our eight G-contours were more effective and more efficient than our depth map method. Our efficient implementation of an ICP based comparison method

achieved a higher performance, but is still less time-efficient than our selected set of contours. This is caused by the iterative selection of corresponding samples during face matching, whereas curve matching allows for the off-line selection of corresponding samples and thus efficient face matching. In SHREC'08, MAPs were achieved ranging from 0.46 up to 0.94 and RRs of 63.5% up to 99.5%. Compared to these results our efficient facial curve based method performs well. For the UND dataset, which contains face scans of higher quality and less facial expressions, our facial curve based method outperforms other methods.

Acknowledgments

This research was partially supported by the FP6 IST Network of Excellence 506766 AIM@SHAPE and FOCUS-K3D FP7-ICT-2007-214993. The authors thank the University of South Florida for providing the USF Human ID 3D Database. We thank Szymon Rusinkiewicz for TriMesh2, Marco Attene for JMeshLib, and VCGLab for their VCG-library, which are very useful libraries to process 3D scan data.

References

- [1] F.R. Al-Osaimi, M. Bennamoun, A. Mian, Integration of local and global geometrical cues for 3D face recognition, *Pattern Recog.* 41 (3) (2008) 1030–1040.
- [2] B. Amberg, R. Knothe, T. Vetter, Expression invariant face recognition with a morphable model, in: *IEEE International Conference on Automatic Face and Gesture Recognition (FG08)*, 2008.
- [3] B.B. Amor, K. Oujii, M. Ardabilian, L. Chen, 3D face recognition by ICP-based shape matching, in: *ICMI*, 2005.
- [4] S. Berretti, A. Del Bimbo, P. Pala, F. Silva Mata, Face recognition by matching 2D and 3D geodesic distances, in: *MCAM07*, 2007, pp. 444–453.
- [5] P.J. Besl, N.D. McKay, A method for registration of 3D shapes, *PAMI* 14 (2) (1992) 239–256.
- [6] V. Blanz, K. Scherbaum, H.-P. Seidel, Fitting a morphable model to 3D scans of faces, in: *ICCV*, 2007, pp. 1–8.
- [7] K.W. Bowyer, K. Chang, P. Flynn, A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition, *CVIU* 101 (1) (2006) 1–15.
- [8] A.M. Bronstein, M.M. Bronstein, R. Kimmel, Three-dimensional face recognition, *IJCV* 64 (1) (2005) 5–30.
- [9] K.I. Chang, K.W. Bowyer, P.J. Flynn, An evaluation of multimodal 2D + 3D face biometrics, *PAMI* 27 (4) (2005) 619–624.
- [10] K.I. Chang, K.W. Bowyer, P.J. Flynn, Multiple nose region matching for 3D face recognition under varying facial expression, *PAMI* 28 (10) (2006) 1695–1700.
- [11] J. Cook, V. Chandran, S. Sridharan, C. Fookes, Face recognition from 3D data using iterative closest point algorithm and Gaussian mixture models, in: *3DPVT*, 2004, pp. 502–509.
- [12] J. Davis, S.R. Marschner, M. Garr, M. Levoy, Filling holes in complex surfaces using volumetric diffusion, in: *3DPVT*, 2002, pp. 428–861.
- [13] T. Faltemier, K. Bowyer, P. Flynn, A region ensemble for 3D face recognition, *IEEE Trans. Inform. Forensics Security* 1 (3) (2008) 62–73.
- [14] B. Gökberk, M.O. Irfanoglu, L. Akarun, 3D shape-based face representation and feature extraction for face recognition, *IVC* 24 (8) (2006) 857–869.
- [15] R. Kimmel, J. Sethian, Computing geodesic paths on manifolds, *Proc. Natl. Acad. Sci.* 95 (15) (1998) 8431–8435.
- [16] C. Li, A. Barreto, J. Zhai, C. Chin, Exploring face recognition by combining 3D profiles and contours, in: *IEEE SoutheastCon*, 2005, pp. 576–579.
- [17] X. Lu, A. Jain, Deformation modeling for robust 3D face matching, *PAMI* 30 (8) (2008) 1346–1356.
- [18] A.S. Mian, M. Bennamoun, R. Owens, Matching tensors for pose invariant automatic 3D face recognition, in: *IEEE A3DISS*, 2005.
- [19] A.S. Mian, M. Bennamoun, R. Owens, Automatic 3D face detection normalization and recognition, in: *DPVT*, 2006, pp. 735–742.
- [20] A. Moreno, A. Sanchez, GavabDB: a 3D face database, in: *Workshop on Biometrics on the Internet COST275*, Vigo, 2004, pp. 77–85.
- [21] P. Nair, A. Cavallaro, SHREC'08 entry: registration and retrieval of 3D faces using a point distribution model, in: *Shape Modeling and Applications (SMI'08)*, 2008, pp. 257–258.
- [22] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, W. Worek, Preliminary face recognition grand challenge results, in: *FGR*, 2006, pp. 15–24.
- [23] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: *3rd International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001, pp. 145–152.
- [24] C. Samir, A. Srivastava, M. Daoudi, Three-dimensional face recognition using shapes of facial curves, *PAMI* 28 (11) (2006) 1858–1863.
- [25] C. Samir, A. Srivastava, M. Daoudi, E. Klassen, An intrinsic framework for analysis of facial surfaces, *IJCV*, (2008), in press.
- [26] A. Scheenstra, A. Ruifrok, R.C. Veltkamp, A survey of 3D face recognition methods, in: *AVBPA*, 2005, pp. 891–899.
- [27] M. Slater, A. Steed, Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison-Wesley Publishers, 2001.
- [28] F.B. ter Haar, M. Daoudi, R.C. Veltkamp, Shape retrieval contest 2008: 3D face scans, in: *Shape Modeling and Applications (SMI)*, 2008, pp. 225–226.
- [29] F.B. ter Haar, R.C. Veltkamp, A 3D face matching framework, in: *Shape Modeling and Applications (SMI)*, 2008, pp. 103–110.
- [30] M. Turk, A. Pentland, Face recognition using eigenfaces, in: *Computer Vision and Pattern Recognition*, 1991, pp. 586–591.
- [31] University of South Florida, Prof. Sudeep Sarkar. USF HumanID 3D Face Database and Morphable Faces, (2003). Available from: <http://www.csee.usf.edu/~sarkar/index_files/DataAndCode.htm>.
- [32] R.C. Veltkamp, F.B. ter Haar, SHREC2007: 3D shape retrieval contest, technical report UU-CS-2007-015, Utrecht University, 2007.
- [33] R.C. Veltkamp, F.B. ter Haar, SHREC2008: 3D shape retrieval contest, in: *Shape Modeling and Applications (SMI)*, 2008, pp. 215–263.
- [34] C. Xu, T. Tan, Y. Wang, L. Quan, Combining local features for robust nose location in 3D facial data, *Pattern Recog. Lett.* 27 (13) (2006) 1487–1494.