

A 3D Face Matching Framework

Frank B. ter Haar*

Remco C. Veltkamp†

Department of Information and Computing Sciences, Utrecht University, the Netherlands



ABSTRACT

Many 3D face matching techniques have been developed to perform face recognition. Among these techniques are variants of 3D facial curve matching, which are techniques that 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 is still under-exposed. 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 literature, and select subsets of facial curves for effective and efficient face matching. Results on the 3D face retrieval track of SHREC’07 (the 3D SHape Retrieval Contest) shows the highest mean average precision achieved so far, using only three facial curves of 45 samples each.

Index Terms: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Retrieval models;

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.

Related work. The task to recognize 3D faces has been approached with many different techniques as described in surveys of Bowyer et al. [5] and Scheenstra et al. [15]. Many methods focus on recognizing 3D faces with neutral expressions, which is still an active field of research. Recently, Al-Osaimi et al. [1] developed a method that combines local and global geometric information of the face in a 2D histogram, extracts a single feature vector, and performs 3D face matching. More challenging is to recognize faces

under different expressions. In [2, 7, 11], the Iterative Closest Point (ICP) algorithm is applied in combination with a region-based metric to define facial similarity mainly on expression invariant regions of the face. Facial curve based methods use a predefined subset of facial curves that are reasonably robust under facial expressions, as in 3D profile and contour matching by Li et al. [10], Samir et al. [14] and Gökberk et al. [8]. Instead of contour lines, Berretti et al. [3] use iso-geodesic stripes and their spatial relationship to identify faces. Bronstein et al. [6] introduce a bending-invariant representation of the face to deal with facial expressions. Several of these 3D face recognition methods require a reference point such as the tip of the nose and a normalized pose of the face. Xu et al. [22] pointed out that the assumption of a nose tip being the vertex with highest z-value doesn’t hold and proposed a more robust method for its detection. To compare face recognition techniques, Face Recognition Grand Challenge sets [12] are publicly available.

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. Second, we propose a 3D face matching framework to extract and match 3D face curves. Thirdly, we evaluate sets of profiles and contours including those described in literature. Fourthly, we propose new combinations of curves to perform both effective and time efficient face retrieval. One of these combinations with only three curves of 45 face samples, achieved the highest mean average precision (MAP) so far of 0.78 in the 3D face retrieval contest of SHREC’07.

Our face pose normalization (Section 3) 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 5, we combine such samples in profile and contour features and select sets of features for effective and efficient face matching.

2 TRAINING SET

In this work we compare 3D faces generated with the morphable face model, which is a point distribution model (PDM) built from 100 face scans of the USF Human ID 3D Database [19]. 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 the $m = 99$ most important eigenvectors of the PDM. 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

*e-mail: fhaar@cs.uu.nl

†e-mail: Remco.Veltkamp@cs.uu.nl

($\sigma_i^2 = \lambda_i$) 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 triangular surface mesh with proper topology and without holes.

Seven instances of the morphable model are selected as a query (q) and 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.

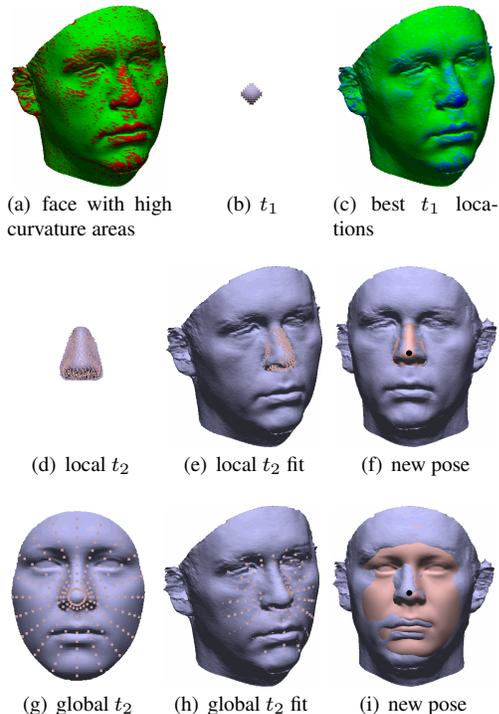


Figure 1: Pose normalization: The original face with high curvature areas (a), the nose tip template t_1 (b), and the optimal (dark blue) locations for t_1 (c). Two different templates t_2 (d,g) fitted to the scan (e,h) to normalize its pose (f,i).

3 POSE NORMALIZATION

Our feature extraction is pose sensitive, so we need to normalize the pose of each 3D face. 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 [16]. 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.

Each vertex of a face model can be considered as a potential nose tip location. Generally, the tip of the nose is a location with high (positive) curvature, which makes it possible to exclude a large number of potential placements based on a simple curvature threshold heuristic. With the use of Rusinkiewicz’s curvature estimation algorithm [13], we obtain an estimation of the mesh’s curvature at

each vertex. We select the vertices with the highest curvature as potential nose tip locations (Figure 1a). Note that these locations include areas around the ears, eyes, lips and chin as well. After that, we apply 3D template matching using a nose tip template t_1 (Figure 1b) to determine which high curvature vertices locally resemble a nose tip (dark blue areas in Figure 1c). The locations where t_1 fits well, we fit a larger template t_2 (as in Figure 1d or g) 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.

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 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 [4], 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 .

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 used during profile extraction.

In this work we use two different templates for t_2 . One uses *local* face information and the other *global* face information of the mean face \bar{S} (Section 2). The local template has samples of the mean face’s nose, while the global template consists of face samples of the entire face. We assume that for each individual, these templates have an 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. Throughout this work we mainly use the local nose template to normalize the face’s pose and to extract the nose tip, which we refer to as *nose detection*. The global face template is used in Section 6, which we refer to as *face detection*. The latter is used to relate our pose normalization to the performance of face retrieval. In Figure 1, a face model is pose normalized twice using these two templates.

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” [10, 14], 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).

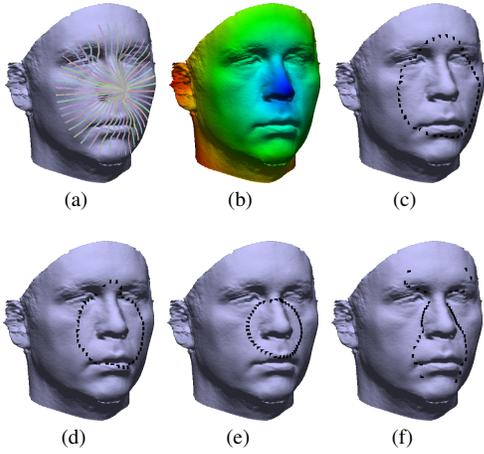


Figure 2: From each face we extract N_p profile curves (a). A G-contour (c) is formed by selecting a sample on each profile (a) that has the same geodesic distance to the tip of the nose (colored in b). form a G-contour (c). Other contour curves are the C-contour (d), XY-contour (e), and Z-contour (f).

To compute the similarity of two faces \mathbf{A} and \mathbf{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 $N \leq N_p \cdot N_c$ samples correspond for faces \mathbf{A} and \mathbf{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

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

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 $d_s(\mathbf{A}_{ij}, \mathbf{B}_{ij})$.

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 5, 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. To crop the face, 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 the chance of missing data or hair covering parts of the

face increases. 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, 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 an approximated 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 [9] 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 Figure 2, the four different contour curves are shown.

Several of the contour and profile curves are used in literature to perform 3D face recognition with. For example, in the work of Li et al. [10] a Z-contour and a XY-profile are used to match 3D faces, Samir et al. [14] use Z-contours, Berretti et al. [3] use stripes of C-contours, and Bronstein et al. [6] use geodesics to create canonical face forms.

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 \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:

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

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.

5 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 evaluate selected sets of features, we use them to query our training set. For each query, a ranked list for the 176 face models is generated, for which we compute the average precision. Then, the *mean average precision* (MAP) over all queries is used to assess the selected features. Note that we aim at the retrieval of relevant faces and not at face identification, which are ranked-first results. In the following paragraphs we determine specific subsets of contours and profiles to perform both effective and efficient face retrieval.

5.1 Single curve matching

For efficient face matching our framework can extract a single contour, 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. Figure 3 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).
- *nd_r* - matching randomly rotated query and original database faces after automatic nose detection.

From these results we learn that,

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.
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 *nd_o* results are comparable to methods that assume scans to be faced forward. For 3D or 2.5D face retrieval the *nd_r* results are important, because this involves pose normalization of faces (or head models) under all possible orientations. For more information about the performance of single curves, we refer to [18].

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 can cause a large decrease in performance. In other words, effective face retrieval based on a single contour curve has a small chance of success. In this section, we assess face matching using multiple curves, based on the *basic*, *nd_o*, and *nd_r* results.

5.2.1 Uniform selection of curves

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 360-360 surface samples exceeds the number of vertices in our face models.

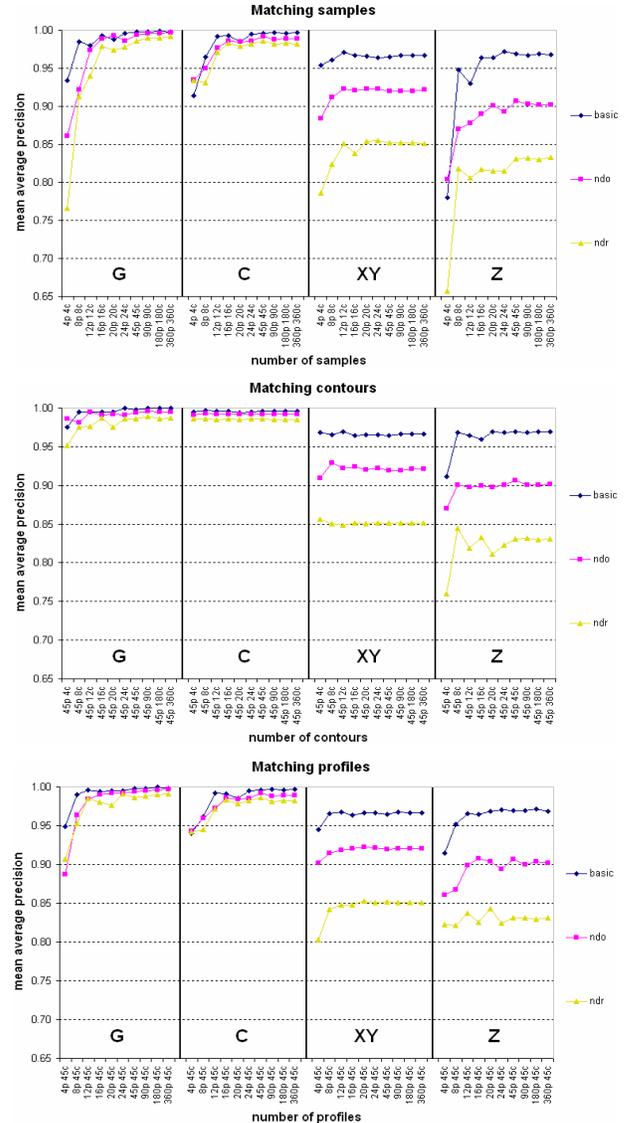


Figure 4: The performance while varying the number of samples (top) contours (middle) or profiles (bottom).

From the results in Figure 4 we learn that the number of samples can be reduced from 360-360 to 45-45 without losing discriminative power. Compared to the number of vertices of a face model, 45-45 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*. Figure 4 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.

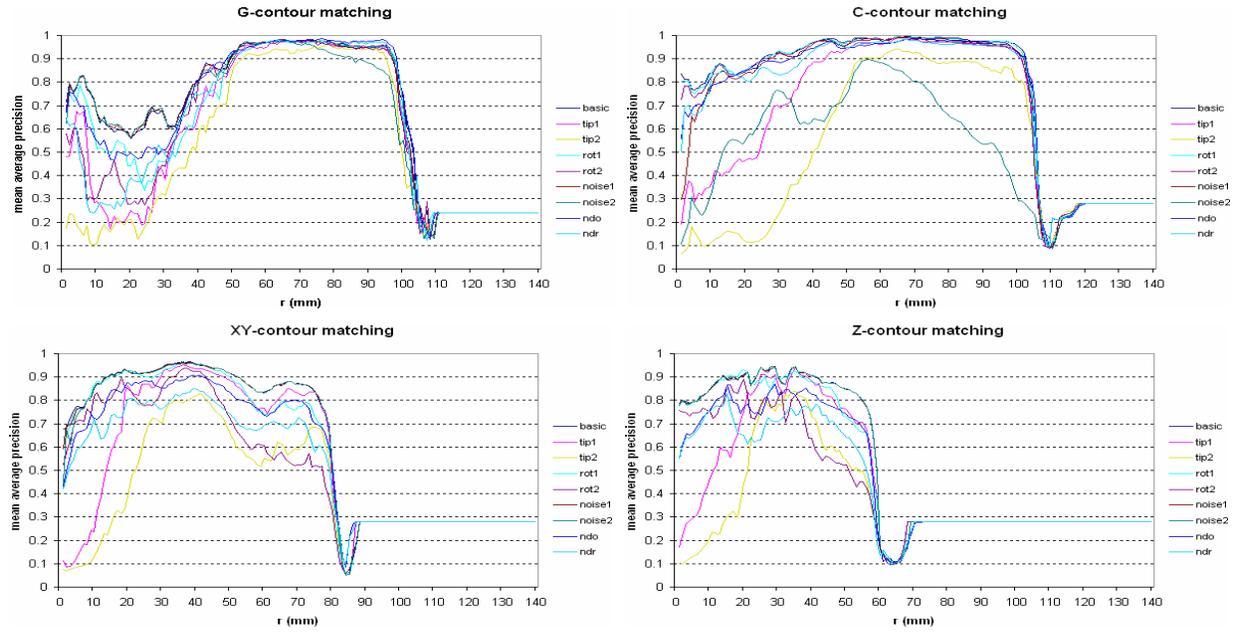


Figure 3: The mean average precision graphs of single G-, C-, XY-, and Z-contours for sample values 1 to 140 mm under varying conditions.

5.2.2 Combination of three curves

In the work of Li et al. [10] face recognition is performed on 2D depth images using the combination of a single Z-contour at distance $z=30\text{mm}$ 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 our training set. 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 Figure 3): the G-contour at 77mm, the C-contour at 68mm, the XY-contour at 36mm, and the Z-contour at 35mm. These contour curves are shown in Figure 2c-f.

Contours and profiles. In this section we explore the 16 hybrid combinations of G,C,XY,Z-contours with $N_p=45$ and G,C,XY,Z-profiles with $N_c=45$. Results on the training set (Figure 5) show a high performance for the combinations of G,C,XY-profiles with G,C,XY-contours. 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.

Contours only. From the previous section, 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. Figure 6 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.

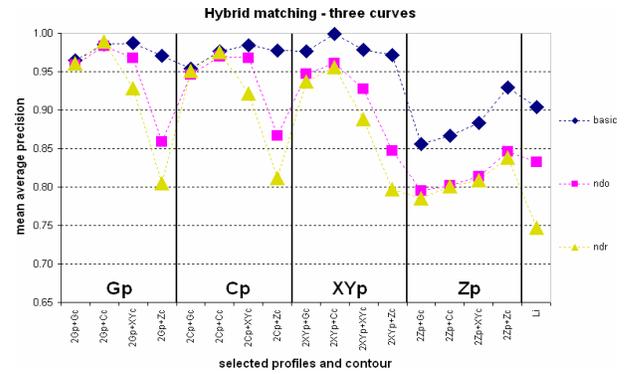


Figure 5: Hybrid matching using one optimally selected contour and two profiles.

5.2.3 Combination of eight curves

From the results in Section 5.2.1 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 5.2.2. 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 Figure 8). These G,C,XY,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. Figure 7 shows the results of the ten unique combinations of G,C,XY,Z-contours. The

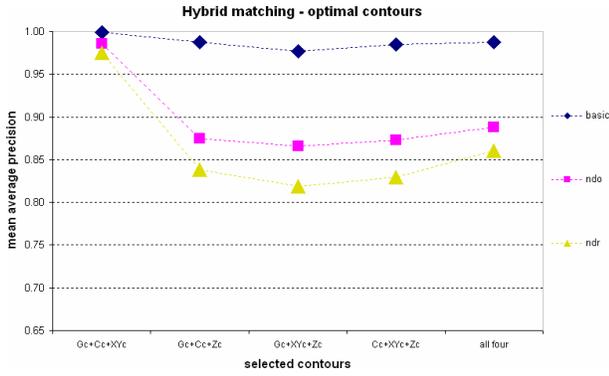


Figure 6: Hybrid matching using the optimally selected contours.

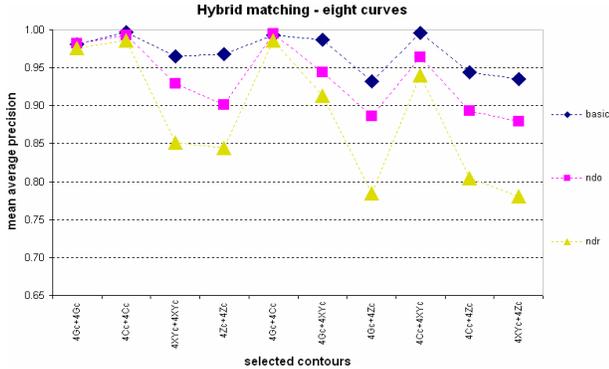


Figure 7: Hybrid matching using combinations of equally spaced contours.

results from this experiment show a high performance for sets of eight G-contours, eight C-contours, and the combination of four G-contours and four C-contours. Adequate results are obtained for combinations of XY-contours with either G- or C-contours.

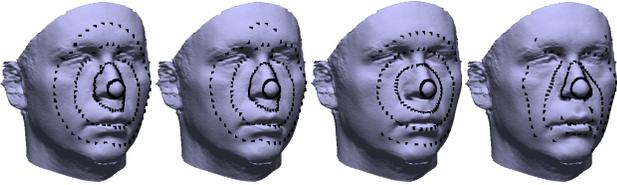


Figure 8: Features used in hybrid matching. From left to right: four G-, C-, XY-, and Z-contours.

6 RESULTS

In this section we compare several settings of our framework on a test set. As a test set we have used the database from “SHREC’07 - Shape Retrieval Contest of 3D Face Models” [21], that consists of 64 queries and 1516 face instances. For the contest, all of the faces were randomly rotated to introduce a non-trivial pose normalization problem. For each of the 64 queries, we query the database and compute the average precision of highly relevant faces. The MAP over all 64 queries is used to evaluate our framework’s settings. The settings we experimented with are those from hybrid matching using three optimally selected curves (Section 5.2.2) and eight uniformly selected curves (Section 5.2.3). For each set of features we show its *basic* and *nd_r* results. To obtain the *basic* results,

we used the ground truth information of this dataset to undo the applied rotations and to select the predefined nose tip locations. These results indicate the optimal performance that can be reached when the pose and nose tip are known. The *nd_r* results were obtained by applying nose detection to faces of the dataset directly, clearly these results are comparable to the results of SHREC’07.

The results on the test set are shown in Figure 9. Observations that count for all results are: (1) The larger embedding of the test set and the greater dissimilarity of relevant faces decreases the overall mean average precision from around 0.9 for the training set to around 0.7 for the test set. (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 global face template instead of the local face template and more view up vectors (every ten degrees) as described in Section 3. 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. The latter resulted in better pose normalization of the faces and a more accurate localized nose tip, which has a positive effect on the retrieval performance (see Figure 9).

6.1 Combination of three curves

For feature sets of three curves, Figure 9 shows good results for the optimally selected G-, C-, and XY-contour in combination with the central XY-profile. Remarkable is that the G- and C-profiles show a relatively large decrease in performance compared to the results from the training set. The C-profile performed very well on the training set because all models had a similar level of noise. The test set on the other hand, contains relevant classified faces which were morphed towards and away from the mean face introducing different levels of noise. This property of the test set 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 $r=0\text{mm}$ to $r=6\text{mm}$, are not effective to retrieve relevant faces (see Figure 3). However, a G-profile contains samples within this range, making the profile curve less reliable. For the rather small training set the central G-profiles were discriminative enough, but the larger embedding of the test set 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 profile 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*.

The results for optimally selected contours on the test set are similar to those on the training set (see Figure 9). 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*.

6.2 Combination of eight curves

For feature sets of eight uniformly selected contour curves, results on the test set (Figure 9) 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 doesn’t 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.

The best results for the test set are 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.

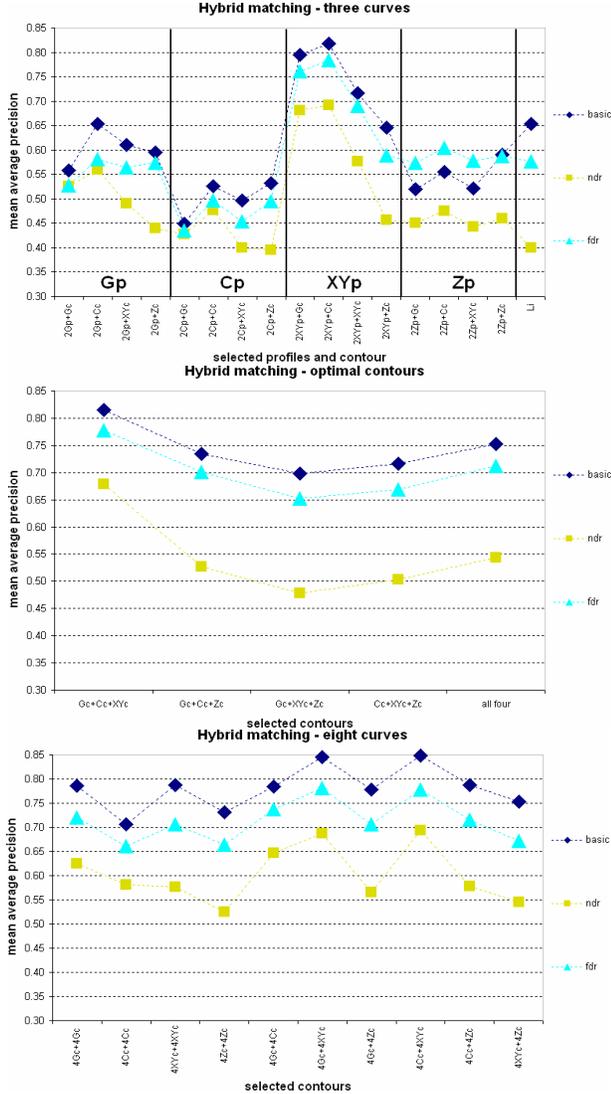


Figure 9: Retrieval results on the test set. The framework was applied using combinations of manually selected curves (top and middle), and eight uniformly selected curves (bottom).

6.3 Comparison to SHREC'07 results

The test set was used in the 3D face retrieval track of SHREC'07. We can compare our obtained nd_r and fd_r results with the MAP results of *highly relevant faces* (MAPH) from this contest. The highest MAPH reported in this contest was MAPH=0.66 using our framework without prior evaluation of its settings. The second best result of MAPH=0.62 was obtained using an ICP-based method using approximately 7600 samples. Now, with an improved pose normalization method and a small number of distinctive face features, we achieve higher mean average precisions of up to 0.78 (see Table

features	#samples	nd_r	fd_r	<i>basic</i>
ICP	7,600	0.62	-	-
180Cp	9,000	0.66	-	-
180XYp	9,000	0.56	-	-
180Zp	9,000	0.59	-	-
Li	135	0.40	0.58	0.65
2XYp + Gc	135	0.68	0.76	0.79
2XYp + Cc	135	0.69	0.78	0.82
Gc + Cc + XYc	135	0.68	0.78	0.82
4XYc + 4Gc	360	0.69	0.78	0.85
4XYc + 4Cc	360	0.69	0.78	0.85

Table 1: The MAPH results on the SHREC'07 test set.

1). The figure below the title shows the first six items of a query's ranked list, see [20] for more results.

6.4 Discussion

The representation of a 3D face as a set of selected facial curves, results in a large reduction of data, which enables efficient face matching. This data reduction can be achieved similarly using an ICP-based method that selects correspondences for a small subset of vertices. However, curve matching allows for off-line feature extraction and thus fast face matching using the predefined corresponding samples. An ICP-based method on the other hand, iteratively determines the set of corresponding samples during face matching. To deal with this inefficiency, Mian et al. [11] proposed to apply a low cost rejection classifier that eliminates a large percentage of potential face matches, before the actual face matching.

The test set we used in this work is considered to be a difficult one. Besides the non-trivial problem of pose normalizing the 3D face models, the large embedding of this dataset and the way relevant faces were generated makes it hard to retrieve the faces that are classified as relevant. Throughout this work, relevant faces were created by morphing the query towards an other instance of the morphable model. In our opinion all created instances resembled their query, even when we applied a 50-50 weighting scheme between two very different faces. However, whether or not these synthetically generated faces are truly relevant to their query is subjective.

As stated earlier, the test set consists of instances of the morphable face model that could be considered synthetic data. However, the morphable face model is a statistical model of real 3D face scans able to generate new faces with a quality similar to these scans. Because recent research shows that the morphable face model can be fitted to 3D scan data, we can approximate other 3D face scans with the optimally generated instance of the face model. Using these approximations of the 3D face data, we can perform 3D face recognition with the characteristic curves of the human face selected in this work.

An important aspect of 3D face matching is the handling of facial expressions. For example, in [14] a subset of Z-contours were selected that are reasonably robust for a dataset of six different expressions per person. In the previous section, we used our framework to select subsets of contours that can be used for effective and efficient face matching under facial morphing. Similarly, we could use our framework to select the optimal set of contours for expression invariant face matching.

In [11] an ICP variant is applied to a masked face region (nose, eyes and forehead) which is assumed to be static under facial expressions. With our framework we can simply extract and match the profiles within this region to obtain a "masked" region. Although several expressions are evaluated in their work, the assumption that a face has a region that remains static under all possible muscle contractions doesn't hold. Moreover, the facial expressiveness varies for different people. Instead of restricting our framework to a specified subset of profiles, we could 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. In SHREC'08 [17] we show that the selection of best matching profiles increases the retrieval performance in the presence of facial expressions.

6.5 Timings

The implementation of our 3D face matching framework requires 40 milliseconds to extract $N_p \cdot N_c=45 \cdot 45$ samples from a face. The extraction of G-samples takes more time to compute the mesh's geodesics. To query our training set of 176 faces using $N_p \cdot N_c=45 \cdot 45$ samples per face, our framework takes nearly 3.5 seconds. This is approximately 20 milliseconds for the matching of two faces using 2,025 samples. The time required for the matching linearly depends on the number of extracted samples. The pose normalization needs 15 seconds. All timings are based on a Pentium IV 2,8 GHz.

7 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. For a fair comparison of facial curves a generic distance measure was used.

With our proposed framework, we examined the properties of profile and contour sampling, the performance of single facial curves, uniform selected curves, manually selected curves, and hybrid combinations. The main results are:

1. C-contours are more robust to errors in nose tip localization and pose normalization, and XY-, and Z-contours are more robust to noise.
2. G-contours on the outer regions of the face are robust under pose, noise and nose tip changes.
3. A few number of contours are more effective than a few number of profiles.
4. Effective face matching requires multiple curves.
5. Efficient and effective face matching is possible with eight uniformly sampled curves, that is, 360 samples.
6. Hybrid matching can further improve on the effectiveness of face retrieval.
7. Manual selection of fewer curves improves the efficiency even further.

Two different combinations of three curves achieve the highest results of 0.78 on the 3D face retrieval track of SHREC'07, the feature set of two XY-profiles and one C-contour and the feature set of the optimally selected G-, C-, and XY-contour. Both feature sets consist of only 135 samples extracted from the face surface, which allows highly efficient face matching.

So far, we have evaluated several settings of our framework for their effectiveness and efficiency in a 3D face retrieval contest. Our framework allows several ways to match faces with different expressions. In our future work, we will explore the possibilities of our framework to expression invariant face matching. Furthermore, we will fit the morphable face model to scans of the FRGC V2.0 and use our selected sets of profiles and contours to identify 3D faces.

ACKNOWLEDGEMENTS

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 would like to 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, and A. Mian. Integration of local and global geometrical cues for 3D face recognition. *Pattern Recogn.*, 41(3):1030–1040, 2008.
- [2] B. B. Amor, K. Oujji, M. Ardabilian, and L. Chen. 3D Face recognition by ICP-based shape matching. In *ICMI*, 2005.
- [3] S. Berretti, A. Del Bimbo, P. Pala, and F. Silva Mata. Face Recognition by Matching 2D and 3D Geodesic Distances. In *MCAM07*, pages 444–453, 2007.
- [4] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *TPAMI*, 14(2):239–256, 1992.
- [5] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *CVIU*, 101(1):1–15, 2006.
- [6] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *IJCV*, 64(1):5–30, 2005.
- [7] J. Cook, V. Chandran, S. Sridharan, and C. Fookes. Face recognition from 3D data using Iterative Closest Point algorithm and Gaussian mixture models. In *3DPVT*, pages 502–509, 2004.
- [8] B. Gökberk, M. O. Irfanoglu, and L. Akarun. 3D Shape-based Face Representation and Feature Extraction for Face Recognition. *IVC*, 24(8):857–869, 2006.
- [9] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds. In *Proc. of National Academy of Sciences*, volume 95(15), pages 8431–8435, 1998.
- [10] C. Li, A. Barreto, J. Zhai, and C. Chin. Exploring face recognition by combining 3D profiles and contours. In *IEEE SoutheastCon*, pages 576–579, 2005.
- [11] A. S. Mian, M. Bennamoun, and R. Owens. Automatic 3D face detection normalization and recognition. In *3DPVT*, pages 735–742, 2006.
- [12] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, and W. Worek. Preliminary Face Recognition Grand Challenge Results. In *FGR*, pages 15–24, 2006.
- [13] S. Rusinkiewicz. Estimating Curvatures and Their Derivatives on Triangle Meshes. In *3DPVT*, pages 486–493, 2004.
- [14] C. Samir, A. Srivastava, and M. Daoudi. Three-Dimensional Face Recognition Using Shapes of Facial Curves. *TPAMI*, 28(11):1858–1863, 2006.
- [15] A. Scheenstra, A. Ruifrok, and R. C. Veltkamp. A Survey of 3D Face Recognition Methods. In *AVBPA*, pages 891–899, 2005.
- [16] M. Slater, A. Steed, and Y. Chrysanthou. *Computer Graphics and Virtual Environments: From Realism to Real - Time*. Addison-Wesley Publishers, 2001.
- [17] F. B. ter Haar and R. C. Veltkamp. SHREC'08 Entry: 3D Face Recognition using Facial Contour Curves. In *These proceedings*.
- [18] F. B. ter Haar and R. C. Veltkamp. A 3D Face Matching Framework. Technical Report UU-CS-2007-047, Utrecht University, 2007.
- [19] University of South Florida, Prof. Sudeep Sarkar. USF HumanID 3D Face Database.
- [20] Utrecht University. SHREC 2007 - Shape Retrieval Contest of 3D Face Models at <http://give-lab.cs.uu.nl/shrec/shrec2007>, Apr 2007.
- [21] R. C. Veltkamp and F. B. ter Haar. SHREC2007: 3D Shape Retrieval Contest. Technical Report UU-CS-2007-015, Utrecht University, 2007.
- [22] C. Xu, T. Tan, Y. Wang, and L. Quan. Combining local features for robust nose location in 3D facial data. *Pattern Recogn. Lett.*, 27(13):1487–1494, 2006.