# A 3D Face Matching Framework

*F. B. ter Haar*

*R. C. Veltkamp*

institute of information and computing sciences, utrecht university

technical report UU-CS-2007-047

www.cs.uu.nl

# A 3D Face Matching Framework

Frank B. ter Haar and Remco C. Veltkamp
Department of Information and Computing Sciences
Utrecht University, the Netherlands

January 29, 2008

### 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 eight facial curves of 45 samples each.

## 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. [6] 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, 8, 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. [9]. Instead of contour lines, Berretti et al. [3] use iso-geodetic stripes and their spatial relationship to identify faces. Bronstein et al. [7] 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. [20] 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 eight curves of 45 face samples achieved the highest mean average precision (MAP) so far of 0.79 on the 3D face retrieval contest of SHREC'07.

Our face pose normalization (Section 3) fits 3D templates on 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. In Section 6 we show how our framework can be extended to match faces under facial changes (e.g. expressions).

# 2 Training set

In this work we compare 3D faces generated with the morphable face model [5] of the USF Human ID 3D Database, which is a point distribution model built from 100 face scans. 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 $S = (x_1, y_1, z_1, ..., x_n, y_n, z_n)^T \in \Re^{3n}$ 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 point distribution model. The mean face $\bar{S}$, the eigenvectors $s_i = (\Delta x_1, \Delta y_1, \Delta z_1, ..., \Delta x_n, \Delta y_n, \Delta z_n)^T$, the eigenvalues $\lambda_i$ ($\sigma_i^2 = \lambda_i$) 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].

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.

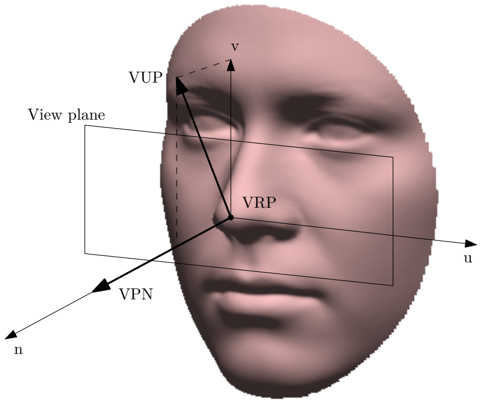
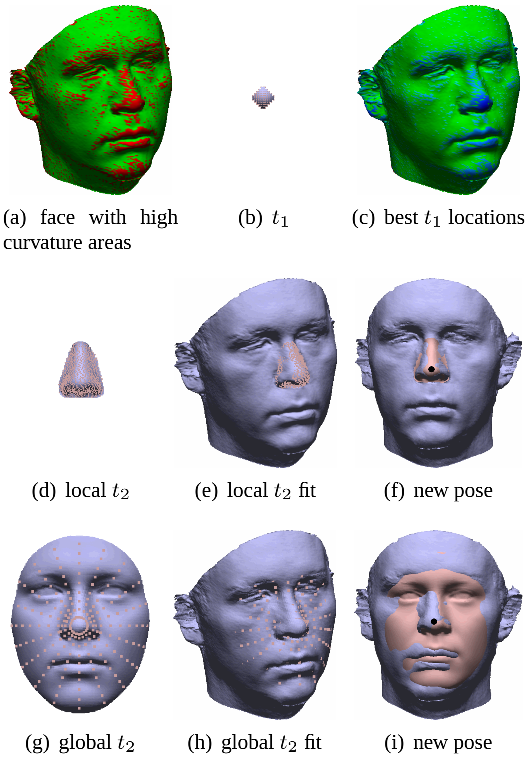


Figure 1: With the nose tip as view reference point (VRP), the gaze direction as view plane normal (VPN), and the face's pose as view up vector (VUP) we obtain the viewing coordinate system to normalize the face's pose.

# 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 (Figure 1) 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 2a). 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 2b) to determine which high curvature vertices locally resemble a nose tip (dark blue areas in Figure 2c). The locations where $t_1$ fits well, we fit a larger template $t_2$ (as in Figure


(a) face with high curvature areas

(b) $t_1$

(c) best $t_1$ locations

(d) local $t_2$

(e) local $t_2$ fit

(f) new pose

(g) global $t_2$

(h) global $t_2$ fit

(i) new pose

Figure 2: 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).

2d 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 returns the Minimum Least Squares (MLS) 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 $\theta_{t_2}$ around $n_{t_2}$. Because the ICP algorithm is able to correct for small rotations we experimented with a subset of only eight different view up vectors (i.e. $\theta_{t_2}$). Each placement of $t_2$ is refined using ICP and the alignment with the lowest MLS 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. 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 7, which we refer to as *face detection*. The latter is used to relate our pose normalization to the performance of face retrieval. In Figure 2, a face model is pose normalized twice using these two templates. Figure 3 shows the pose normalization of four faces using these templates.

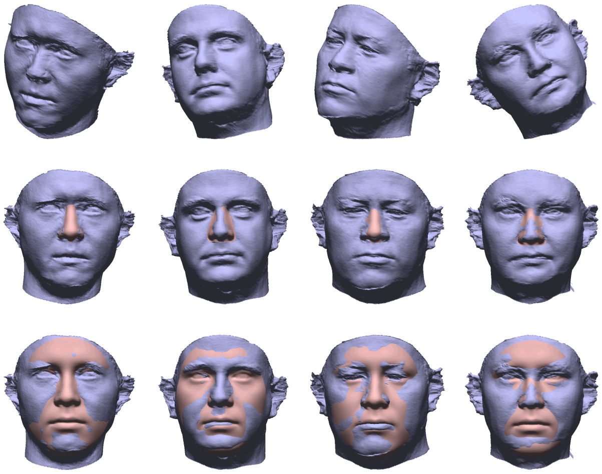


Figure 3: Pose normalization of different faces. The initial faces (top) are pose normalized using the local nose template (middle) and the global face template (bottom).

# 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. Using these sets of profile curves, a similarity of two faces is determined. 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).

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

Our framework extracts a set of $N_p$ profiles. We define a profile as a 3D curve that starts from the tip of the nose and follows a path over the surface mesh with a selected 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

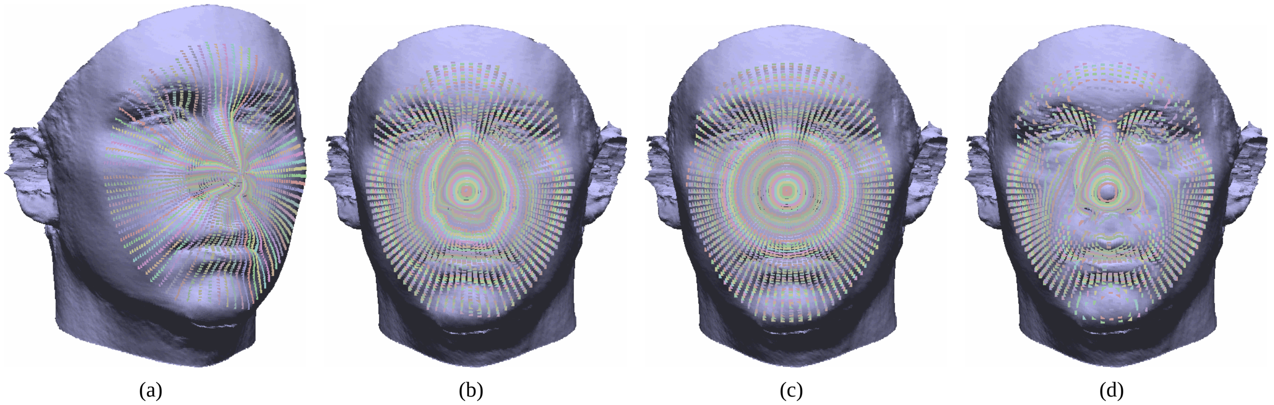


Figure 4: A set of 90 profiles (a) are sampled using increasing C-distance (b), XY-distance (c), and Z-distance (d) resulting in different contour curves. Samples with the same color belong to the same curve.

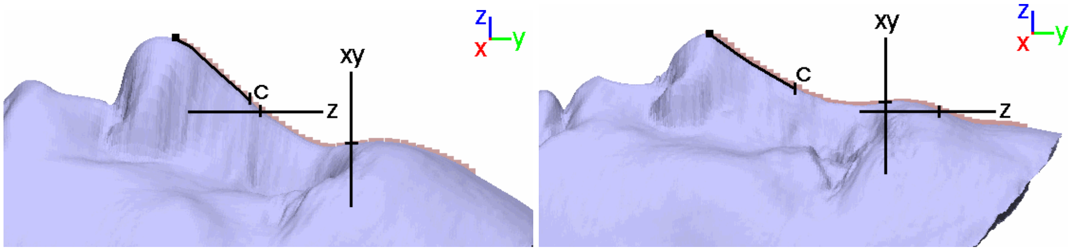


Figure 5: The central profile of two different faces with a "corresponding" C-, XY- and Z- sample.

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

A single profile is a 3D curve over the surface starting at the origin. After normalization, 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 $\mathbf{B}_{ij}$). In this work we specify three different contour samples:

- Samples with a curve distance of $c = r$ over the profile curve to the origin, called the *C-samples*.
- Samples with a $\sqrt{(x^2 + y^2)} = r$ distance to the origin, called the *XY-samples*.
- Samples with a $z = r$ distance to the origin, called the *Z-samples*.

A profile which is sampled with $N_c$ C-samples (with increasing $r$) is referred to as C-profile. A C-contour is the set of $N_p$ C-samples at the same distance $r$. In Figure 4 the three different sampling strategies are shown with the use of contour curves. For the actual face matching we extract a set of at most $N_p \cdot N_c$ samples.

## 4.3 Feature matching

The extracted $N_p \cdot N_c$ samples from one face have an one-to-one correspondence to those of an other face (see Figure 5). To match those samples we apply a symmetric distance measure that can be used for all three 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 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, 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 contour matching under varying conditions that are common in practice.

- The original face instances were used, for which the pose and nose tip are known. We call these $basic$ results.
- The queries were disrupted with two displacements of the nose tip: $tip_1$=2mm and $tip_2$=4mm from the actual nose tip.
- The queries were disrupted with two Euler rotations ($\rho$,$\rho$,$\rho$): $rot_1$ with $\rho$=1 and $rot_2$ with $\rho$=2 degrees.
- The queries were disrupted with additional noise relative to the average edge length $\eta$ in the mesh: $noise_1$ with $0.1\eta$ and $noise_2$ with $0.2\eta$.
- Because the nose tip location is usually unknown, we test the curve matching robustness under automatic nose detection (Section 3) applied to the original database and query faces (called $nd_o$ results) and again for the queries after applying a random rotation (called $nd_r$ results).

The $basic$ results can be used as a reference for the optimal results. A curve is called robust under a varying condition when its performance stays close to the $basic$ 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.

**Single contour**. To investigate how well a single contour can be used for face retrieval, we plot the MAP of each contour type within the range $r$=[1,140] mm. A single contour has $N_p$ samples at a distance $r$. The performance of each single contour is visualized in Figure 6. The most important observation is that C-contours are more robust under nose tip displacements ($tip_1$,$tip_2$) and rotations ($rot_1$,$rot_2$) than the XY- and Z-contours, but less robust under varying noise conditions ($noise_1$,$noise_2$). The nose detection ($nd_o$,$nd_r$) causes both nose tip displacements and small variations in pose, which again results in higher robustness for the C-contours. For the three contour types we selected *active regions* of $r$=1 mm to the $r$ just before the MAP drops to a minimum. The performance drop is due to a lack of samples caused by cropping the face at an Euclidean distance of 90 mm. For the C-contour we selected region $r$=[1,105] mm, for the XY-contour $r$=[1,80] mm, and for the Z-contour $r$=[1,60]mm.

**Single profile**. 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 ($nd_o$,$nd_r$). Figure 6 shows that the maximal performance for a single contour is higher

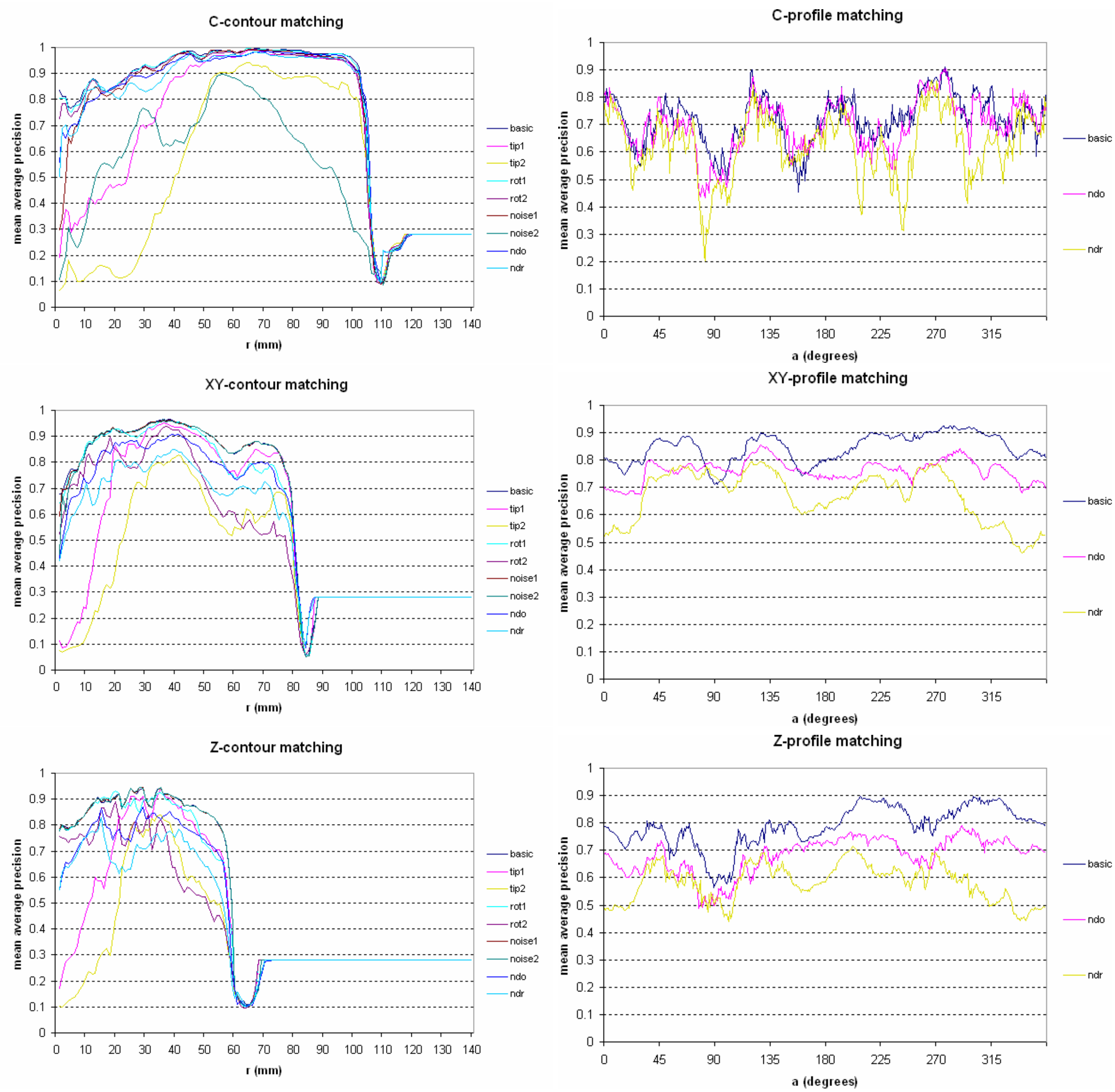


Figure 6: The mean average precision graphs of single C-, XY-, and Z-contours for sample values 1 to 140 mm under different settings (left) and single C-, XY-, and Z-profiles for angles 0 to 359 degrees (right).

than for a single profile, which means that a single contour can be more descriptive than a profile. Remarkable is that the profiles with an angle between 80 and 100 degrees have a lower $basic$ performance for all sampling strategies. These profiles start from the tip of the nose and go upwards to the forehead, which means that this region is less distinctive than other regions in the face.

## 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 performance. In other words, effective face retrieval based on a single contour or profile 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 were 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). Thus the 360$^{th}$ C-contour has samples with $r$=105 mm. 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 samples exceeds the number of vertices in our face models.

From the results in Figure 7 we learn that the number of samples can be reduced from 360·360 to 45·45 without loosing 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 7 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.

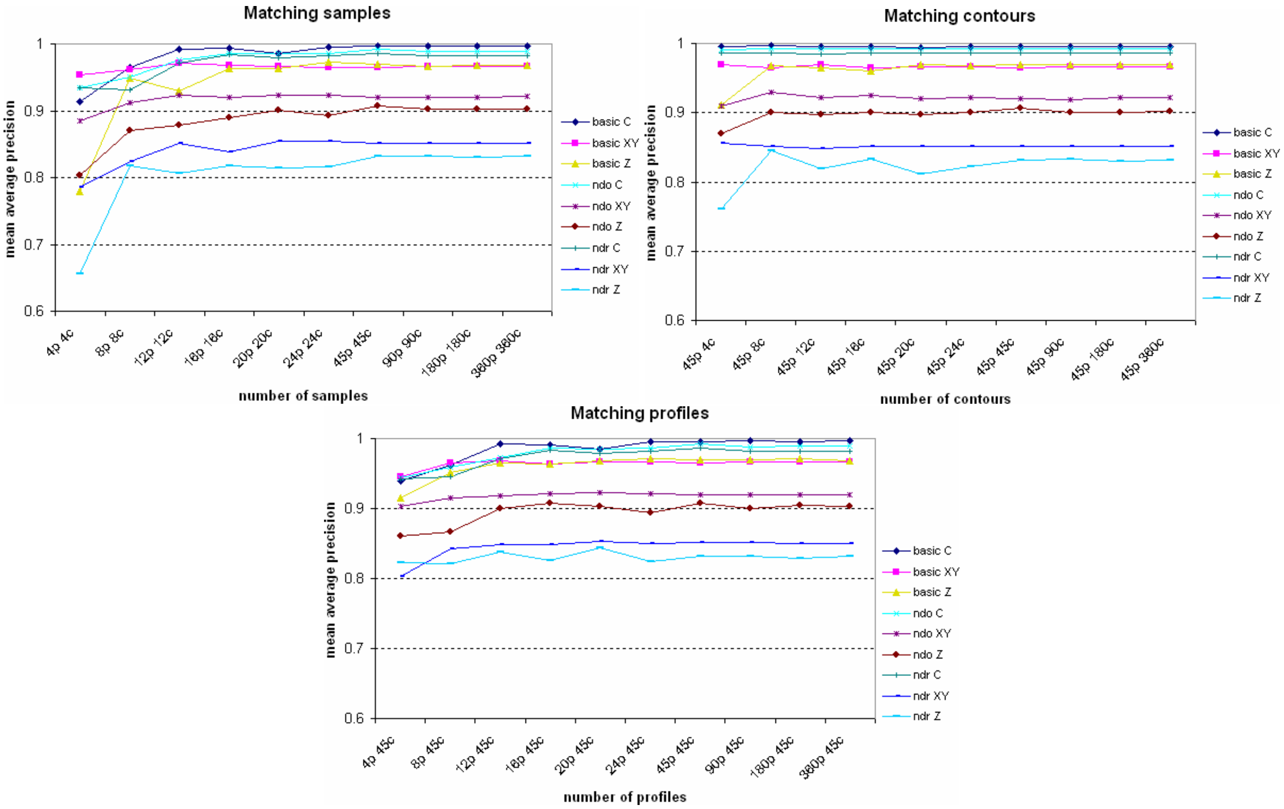


Figure 7: The performance while varying the number of samples (top) contours (middle) or profiles (bottom).

### 5.2.2 Manual selection of curves

To further improve the efficiency of the face retrieval without loosing effectiveness, we manually selected a smaller subset of good performing contours and profiles. Samir et al. [14] for instance obtained a good recognition rate using a subset of one to six Z-contours. We plotted the individual performance of each of the 360 contour curves from the active region based on $N_p$=45 profiles. In Figure 8 we show the MAPs of the $basic$, $nd_o$ and $nd_r$ results and their average. From the average of these results we selected for each of the contour types six peaks with a high performance, starting with the highest peak. The selected indices are shown in Table 1.

For optimal profile matching it is more difficult to select a proper subset, because the graphs in Figure 6 fluctuate much. Furthermore, on average we would expect facial symmetry in the YZ-plane. So we need to find an optimal set of *symmetric profiles*. To do so, we combine the contour samples ($N_c$=45) of two symmetric profiles and plot the performance. This gives us new profile performance graphs for the range $a$=[90,270] (Figure 9). Again we look at the averaged performance of the $basic$, $nd_o$ and $nd_r$ results and select peaks for which face matching has a high performance. Because each peak represents two symmetric profile curves, we selected three peaks which gives us six different profiles in the range $a$=[0,359]. The selected indices are shown in Table 1.

In this experiment we compare the retrieval performance using an increasing number of manually selected profiles and contours. At first we selected the optimal C-contour and determined its retrieval performance, then we added the samples from the second best C-contour from Table 1 and tested their effectiveness, and so on until all six contours were used. This experiment was repeated for the XY-contours and Z-contours and for the profiles as well, obtaining the results in Figure 10. The following observations can be made:

- An optimal contour is more descriptive than an optimal profile.
- The use of multiple curve data often improves the retrieval performance.
- Four manually selected curves slightly improves (on average) the performance of four uniform selected curves (Figure 7).
- Multiple C-curves outperform multiple XY-curves, which have in turn a higher performance than Z-curves.

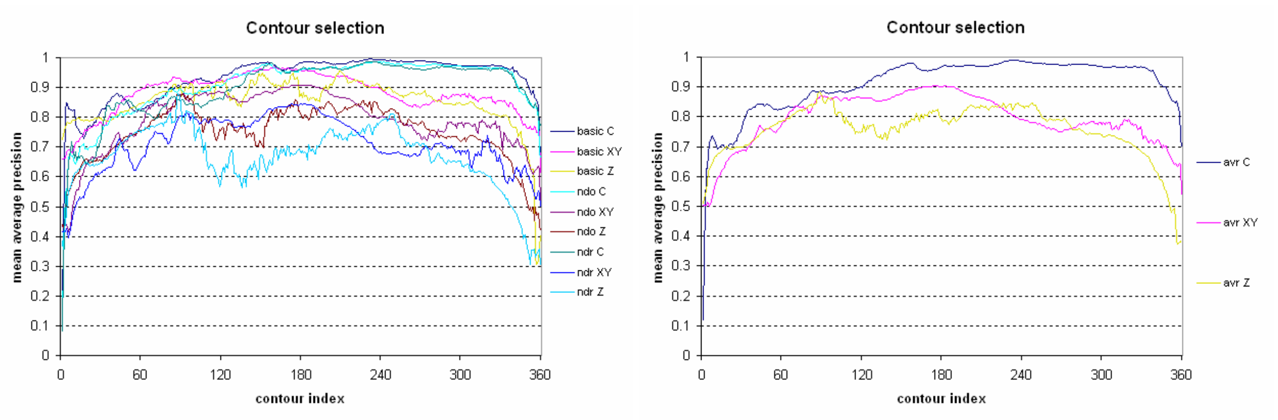


Figure 8: The individual performance of 360 contours equally spaced over the active region of each contour type (left). The average of the $basic$, $nd_o$ and $nd_r$ results per contour type (right).

| Curve | Selected indices (1-360) | | | | | |
|---|---|---|---|---|---|---|
| | 1st | 2nd | 3rd | 4th | 5th | 6th |
| C-contour | 235 | 156 | 281 | 329 | 89 | 45 |
| XY-contour | 173 | 89 | 198 | 116 | 317 | 43 |
| Z-contour | 88 | 237 | 202 | 177 | 272 | 129 |
| C-profile | 264 | 276 | 194 | 346 | 121 | 59 |
| XY-profile | 132 | 48 | 218 | 322 | 237 | 303 |
| Z-profile | 208 | 332 | 231 | 309 | 132 | 48 |

Table 1: Manually selected curves with high performance in descending order.

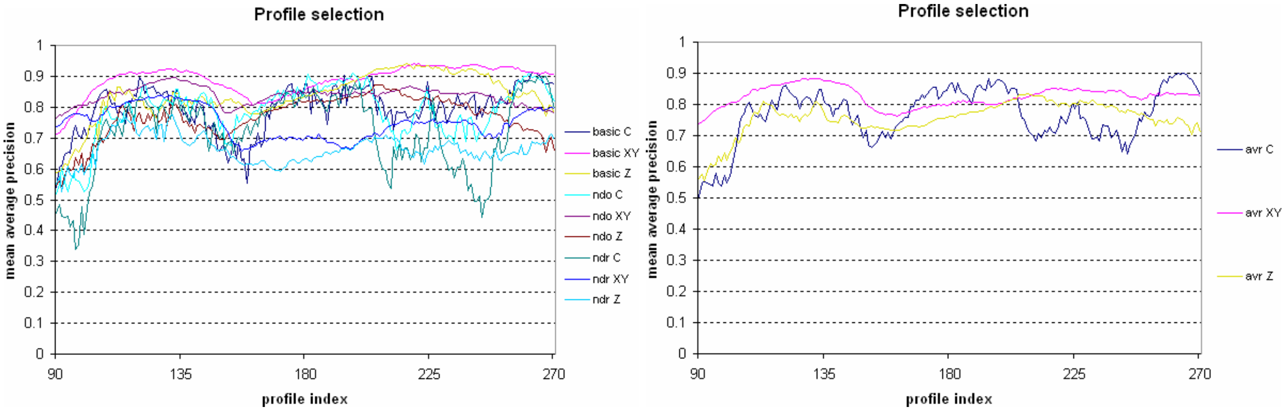


Figure 9: The performance of 180 pairs of symmetric profiles (left) and the average of the $basic$, $nd_o$ and $nd_r$ results (right).

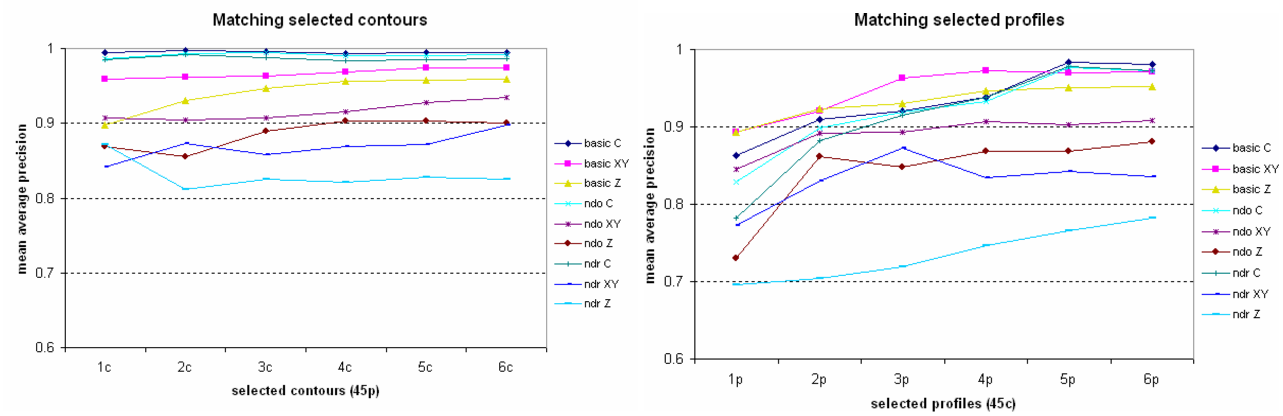


Figure 10: Performance versus the number of manually selected curves.

### 5.2.3 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 $z$=30mm with a vertical profile 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 ($a$=90), and the XY-profile from nose to chin ($a$=270).

Similarly we can combine the *manually selected* contours and profiles among different sampling strategies, which we refer to as *hybrid matching* (see Figure 12). In this section we explore the nine hybrid combinations of C,XY,Z-contours with $N_p$=45 and C,XY,Z-profiles with $N_c$=45. Results on the training set (Figure 11) show a high performance for the combined C-contours and C-profiles. The marked areas show common factors of the results. For the training set the 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 symmetric profiles. Of course the set of three optimal curves may differ per training set.

### 5.2.4 Combination of eight curves

From the results in Section 5.2.1 we have learned that eight 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 (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 5.2.3. In the following experiment we use hybrid combinations of eight curves in an attempt to improve the performance. As far as we know, no-one ever tried the combination of profiles nor contours from different sampling strategies (e.g. C-contours with Z-contours and XY-profiles with C-profiles).

In this experiment we used our framework to generate four equally spaced contours ($N_p$=45) and four equally spaced

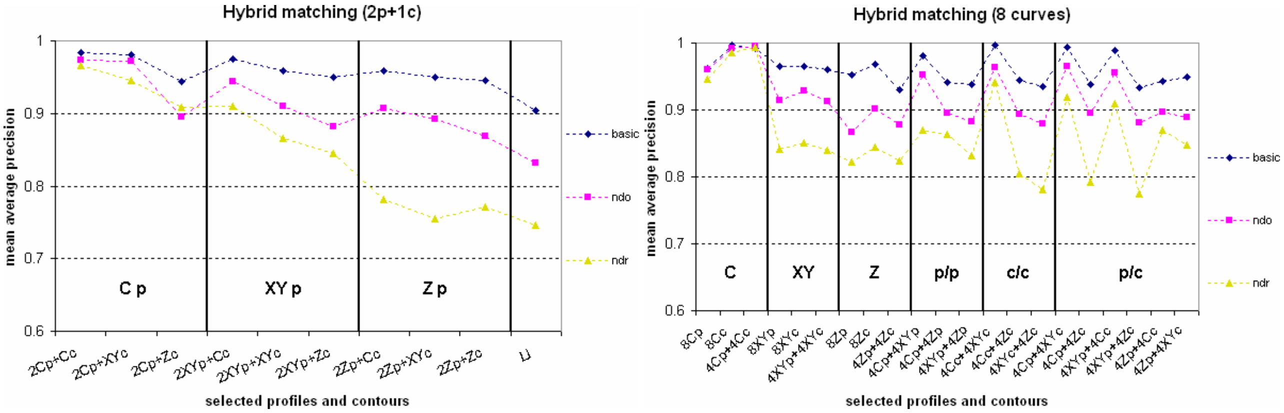


Figure 11: Hybrid matching using two optimally selected profiles and one contour (left) and combinations of equally spaced profiles and contours (right).

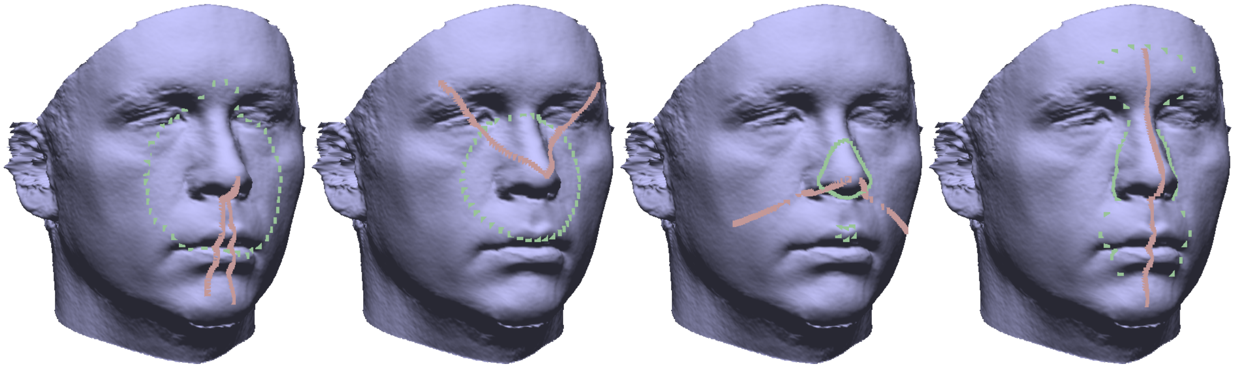


Figure 12: Manually selected curves for hybrid matching. From left to right: one C-contour and two C-profiles, one XY-contour and two XY-profiles, one Z-contours and two Z-profiles, and one Z-contour at depth 30mm and the central XY-profile.

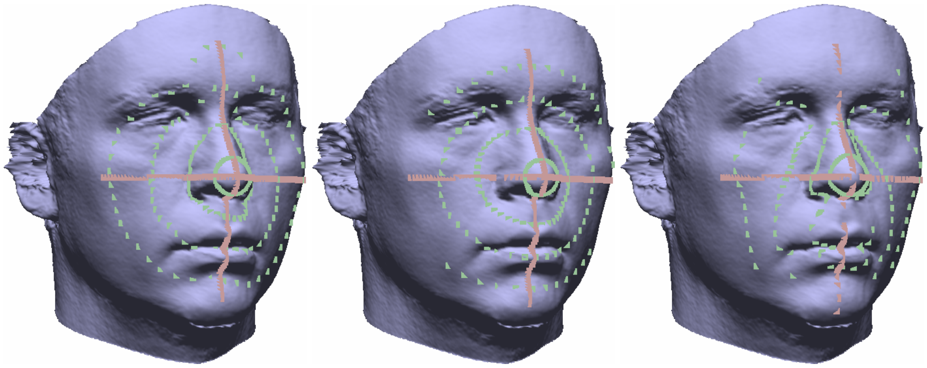


Figure 13: Uniformly selected curves for hybrid matching. From left to right: four C-contours and C-profiles, four XY-contours and XY-profiles, and four Z-contours and Z-profiles.

profiles ($N_c$=45). These features were combined using all possible combinations. Figure 11 shows the results of the 27 combinations: C,XY,Z-contours with C,XY,Z-contours, C,XY,Z-profiles with C,XY,Z-contours, and C,XY,Z-profiles with C,XY,Z-profiles. The samples of three profile-contour combinations are shown in Figure 13. Because combining four C-contours with the same four C-contours (and the five other exact matches) is useless, we use eight equally spaced curves instead. The results from this experiment show minor peaks for eight contour curves of the same type over eight profile curves or a combination of four profiles and four contours of the same type (see areas C, XY and Z). The rest of the graph (areas p/p, c/c and p/c) shows peaks at combinations of XY- and C-curves and valleys when Z-curves are used.

# 6 Facial expressions

Up till now we didn't consider the applicability of our framework to faces with different facial expressions. In previous work, Samir et al. [14] selected a subset of Z-contours that are reasonably robust for their 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.

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. So, we don't want to restrict our framework to a selected subset of profiles. Instead, we select a percentage of profiles that matches best for two input faces. This way, we are even able to recognize a person using profiles that go through the mouth, in case that region remains unchanged from one expression to another.

In this experiment we look at the performance when only a percentage $t_{perc}$ of best matching profiles is used to match faces. Both the reduced number of profiles and contours $N_p \cdot N_c$=45·45 and a larger set of profiles $N_p \cdot N_c$=360·45 are used. Again we evaluate the performance of different contour types, this time while varying the percentage $t_{perc}$={10,20,...,100%}. The retrieval results for our training set (Figure 14) show marginal differences between the use of 45 profiles compared to 360 profiles, which strengthens our applied reduction of facial curves. Furthermore, the $nd_r$ results for the XY- and Z-profiles show that a subset of best matching profiles can improve the performance of contour types that are less robust under nose tip displacements and face rotations. Also, we see that using the best seventy percent of the profiles performs as good as using all profiles. This indicates that a subset of profiles, those that might be disrupted by facial changes, can be neglected without influencing the overall retrieval performance. Note that our training and test set introduce facial changes by morphing the neutral face of one person to another, rather than real facial expressions.

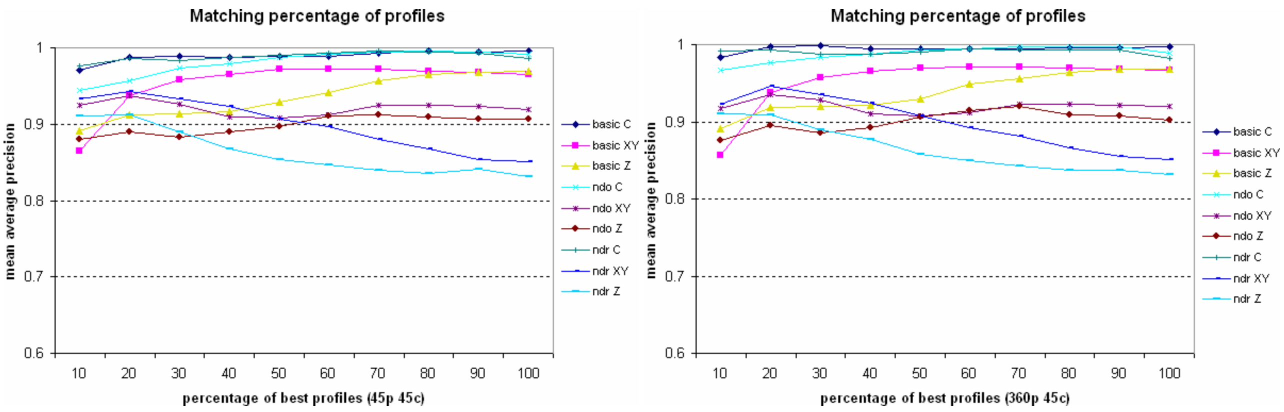


Figure 14: Face matching using percentages of best matching profiles.

# 7 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" [18, 19], 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 and facial expressions:

- Combinations of three curves from Section 5.2.3.
- Combinations of eight curves from Section 5.2.4.
- Percentages of $N_p$=45 profiles using $N_c$=45 contours, as in Section 6.

Again we show for each setting the $basic$, $nd_o$ 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. For the $nd_o$ results, we used the ground truth information of the dataset to undo the applied rotations, and then we applied nose detection to obtain new poses and nose tip locations for all faces. 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 15. Three 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 0.6 for the test set. (2) The performance gap between nose detection applied to rotated faces ($nd_r$) and unrotated faces ($nd_o$) shows that the pose normalization is not accurate enough with eight different view up vectors. (3) 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 third observation we performed an additional experiment using the global face template instead of the local face template and sixteen (instead of eight) different view up vectors as described in Section 3. The global face detection ($fd_r$) resulted in better pose normalization of the faces, which has a positive effect on the retrieval performance (see Figure 15).

Comparing the test results with the training results for combinations of *three curves*, we see a large decrease in performance for the combinations with C-profiles. These combinations 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. Still the combinations with a C-contour perform better than those with a XY-contour, which are in turn better than those with a Z-contour. The best results are obtained using the combinations with two XY-profiles, and in particular the combination of two XY-profiles and one C-contour. The MAP for $nd_r$ in this particular case is 0.62 and for $fd_r$ even 0.72.

With combinations of *eight curves* our framework achieves high results for combinations of XY-curves. The highest performance is obtained for hybrid matching with four C-contours and four XY-contours, followed by the combination of four C-contours with four XY-profiles. With only eight curves of 45 samples, we achieve a MAP of 0.69 for $nd_r$ and 0.79 for $fd_r$. The $basic$ results indicate that the performance can be increased to 0.85, when an even more specialized pose normalization and nose tip localization method is applied.

Various *percentages of best matching profiles* show that the use of best matching profiles instead of all profiles can improve the retrieval performance. The high performance for the XY-profiles for even small percentages shows our framework's potential for datasets with facial changes. Again the $fd_r$ results are better than the $nd_r$ results and closer to the $basic$ results, which shows the effect of an improved pose normalization method to face retrieval.

Comparing the three performance graphs of Figure 15 shows that $N_p \cdot N_c$=45·45 samples are not necessarily better than eight curves of 45 samples. With hybrid matching the effectiveness of eight curves can even be increased, without giving up time efficiency for the face matching. To increase the efficiency even further one can manually select, with some effort, just two or three curves that perform well on a training set. However, the general applicability of such curves to different datasets is not very high: (1) Results from our training set show that C-curves perform best, while results on our test set show the opposite. (2) Li selected an optimal Z-contour and XY-profile based on his training set, but on both our training and test set these curves perform not very well.

## 7.1 Comparison to SHREC'07 results

The test set was used in the 3D face retrieval track of SHREC'07 [18, 19]. 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. This score was set using our pose normalization method using the nose template and our framework without prior evaluation of its settings. We performed highly inefficient matching using $N_p$=180 profiles with $N_c$=50 C-samples and matched multiple combinations of profiles. In the end 270,000 samples were matched per face. The second best result of MAPH=0.62 was obtained using an ICP-based method using approximately 7600 samples.

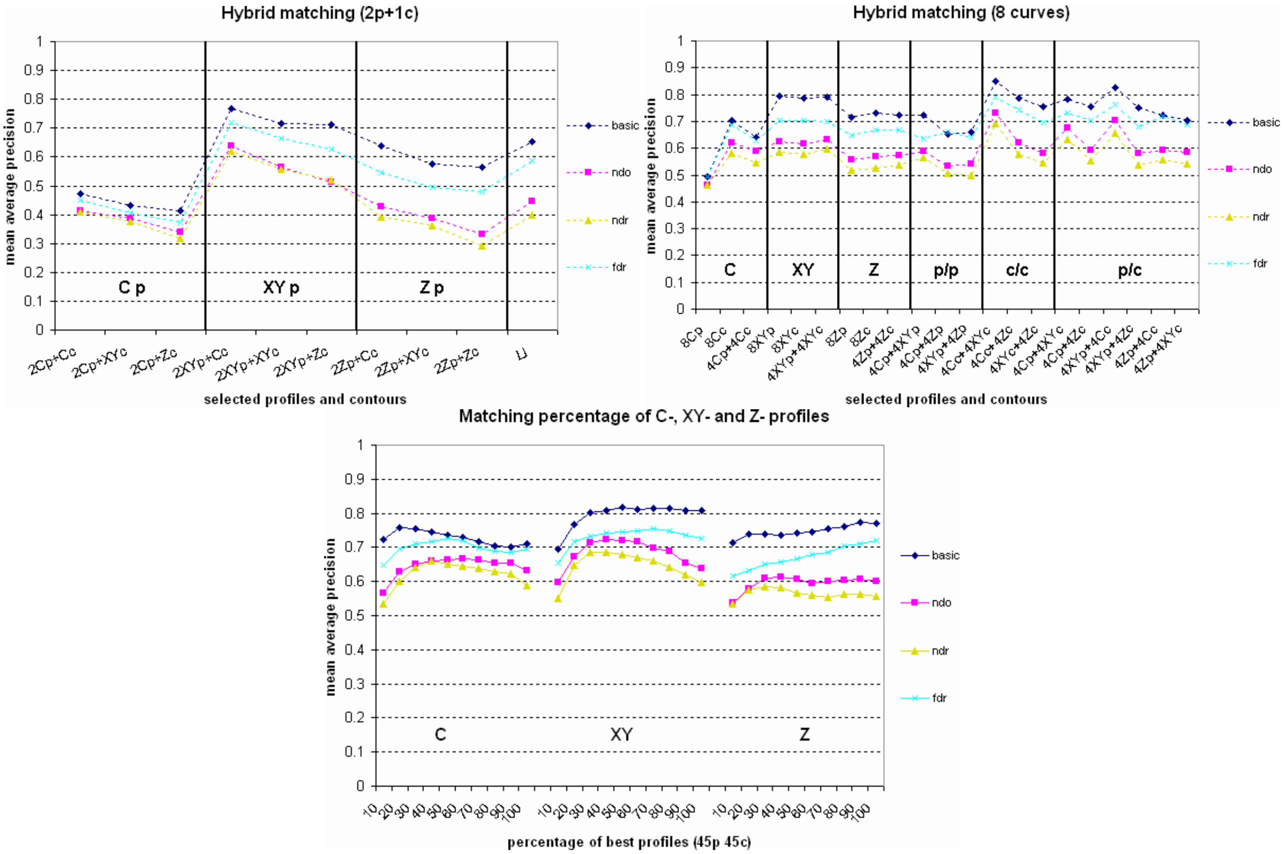


Figure 15: Retrieval results on the test set. The framework was applied using combinations of three curves (top), eight curves (middle) and a percentage of profiles (bottom).

When we take a fraction of the $N_p$=45 profiles with $N_c$=45 (2,025 samples), we achieve higher mean average precisions: 0.72 for 100% of the Z-profiles, 0.73 for 50% of the C-profiles, and 0.76 for 70% of the XY-profiles. The combination of four C-contours with four XY-profiles (only 360 samples shown in Figure 16) performs well with a MAPH of 0.76. The highest MAPH of 0.79 is obtained with four C-contours with four XY-contours. Based on these settings, we show in Figure 17 for five queries the first six items of their ranked lists.

## 7.2 Discussion

The representation of a 3D face using a set of selected facial curves results in a large data reduction that 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 offline 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. Mian et al. [11] reduced the face recognition time of their ICP-based method using a low cost rejection classifier that eliminates a large percentage of potential face matches.

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. We can relate our highest face retrieval performance of MAP=0.79 to a face recognition rate as follows. For 61 of the 64 queries (95,3%) a relevant face (other than the identical face) was found on top of the list. Knowing that a relevant face resembles for 60% the query and for 40% another model, this is a good result.

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. Therefore, we expect that our results apply to other face sets as well.

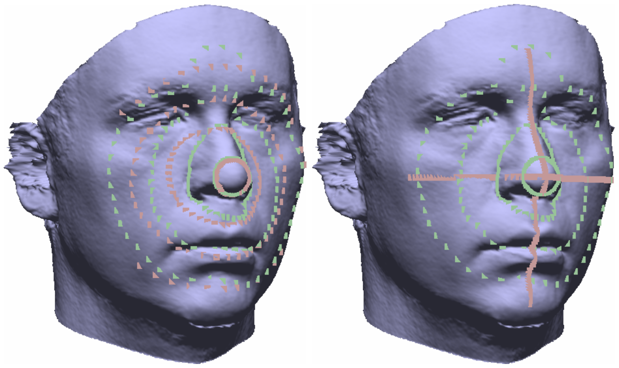


Figure 16: The combination of four C-contours and XY-contours (left) and four C-contours and XY-profiles (right) with high performance on the test set.

### 7.3 Timings

The implementation of our 3D face matching framework requires 40 milliseconds to extract $N_p \cdot N_c$=45·45 samples from a face. To query our training set of 176 faces using $N_p \cdot N_c$=45·45 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 with 520 MB internal memory.

## 8 Concluding remarks

In this work we proposed a new pose normalization technique 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 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-curves are more robust to errors in nose tip localization and pose normalization, and XY-, and Z-curves are more robust to noise. (2) A single contour can be more effective than a single profile. (3) Effective face matching requires multiple curves. (4) Efficient and effective face matching is possible with eight uniform sampled curves, that is, 360 samples. (5) Hybrid matching can further improve the effectiveness. (6) Manual selection of fewer curves improve the efficiency even further at the cost of over-fitting.

The combination of C-contours with XY-profiles, and C-contours with XY-contours achieve the highest results on the 3D face retrieval track of SHREC'07. These features with only 360 face samples in total have a retrieval performance of 0.76 and 0.79, respectively.

Furthermore, we have shown that our face matching framework can handle facial changes in a flexible manner. Using a percentage of best matching profiles our framework automatically selects regions of the face that remain unchanged. Results indicate that only 45 profiles having 45 contour samples each are sufficient, which requires no more than 20 milliseconds to match two faces.

So far, we have evaluated several settings of our framework for their effectiveness and efficiency in a 3D face retrieval contest. In our future work, 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 supported by the FP6 IST Network of Excellence 506766 AIM@SHAPE. The authors thank the University of South Florida for providing the USF Human ID 3D Database.

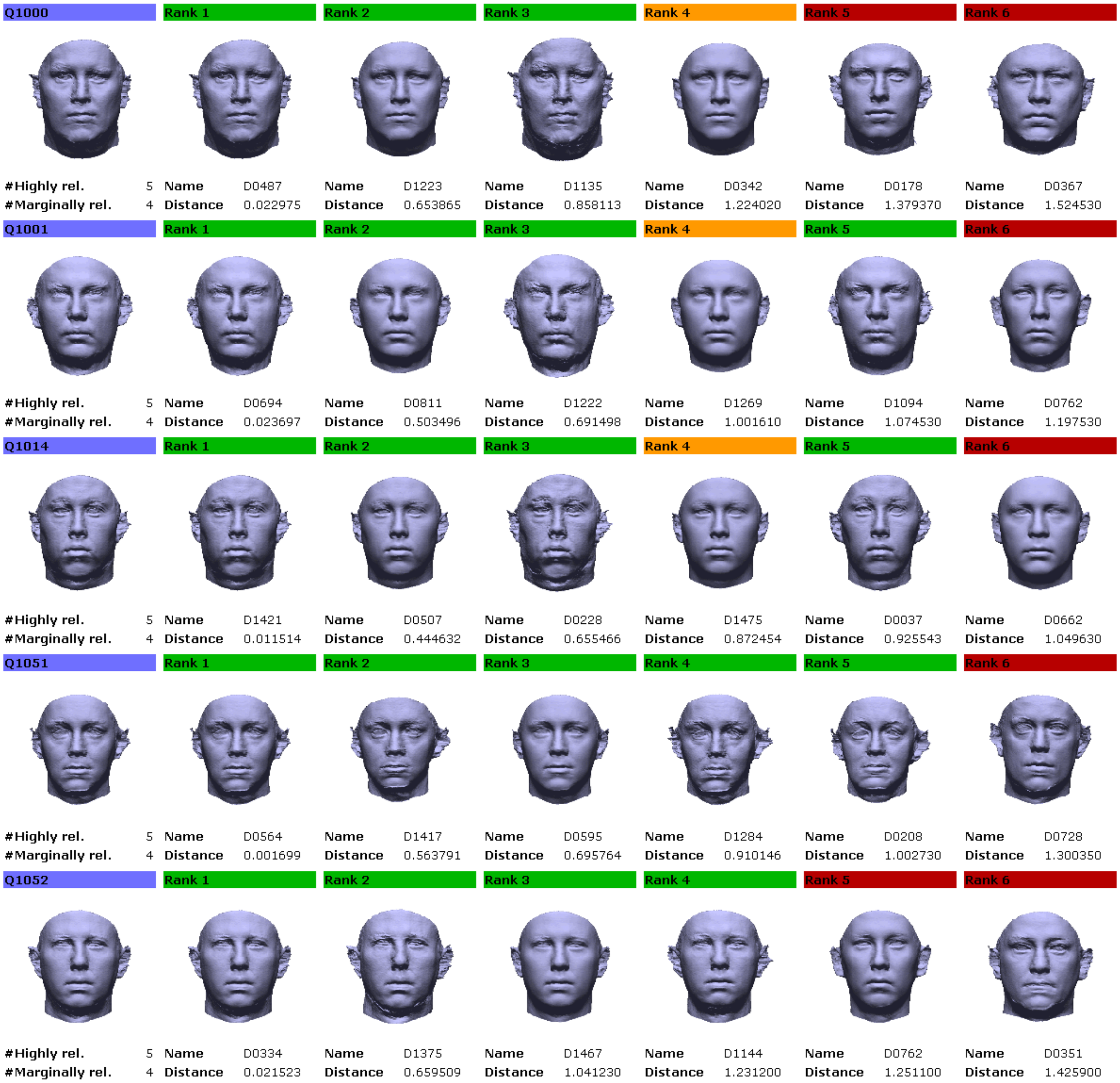


Figure 17: With the combination of four C-contours and XY-contours, several highly relevant items (green) are retrieved on top of the list for these SHREC'07 queries (blue). For more results see [17].

# 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. Ouji, M. Ardabilian, and L. Chen. 3D Face recognition by ICP-based shape matching. In *ICMI*, 2005.

[3] S. Berretti, A. D. Bimbo, and P. Pala. Description and retrieval of 3D face models using iso-geodesic stripes. In *MIR*, pages 13–22, 2006.

[4] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *TPAMI*, 14(2):239–256, 1992.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

[6] 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.

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *IJCV*, 64(1):5–30, 2005.

[8] 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.

[9] 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.

[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. Face retrieval results at `http://ensor.labs.cs.uu.nl/faceretrieval`, Dec 2007.

[18] Utrecht University. SHREC 2007 - Shape Retrieval Contest of 3D Face Models at `http://give-lab.cs.uu.nl/shrec/shrec2007`, Apr 2007.

[19] R. C. Veltkamp and F. B. ter Haar. SHREC2007: 3D Shape Retrieval Contest. Technical Report UU-CS-2007-015, Utrecht University, 2007.

[20] 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.