EG UK Theory and Practice of Computer Graphics (2007)
Ik Soo Lim, David Duce (Editors)

A Statistically-Assisted Sketch-Based Interface for Creating Arbitrary 3D Faces

Orn Gunnarsson and Steve Maddock

The University of Sheffield, England

Abstract

Creating faces is important in a number of application areas. Faces can be constructed using commercial modelling tools, existing faces can be transferred to a digital form using equipment such as laser scanners, and law enforcement agencies use sketch artists and photo-fit software to produce faces of suspects. We present a technique that can create a 3-dimensional head using intuitive, artistic 2-dimensional sketching techniques. Our work involves bringing together two types of graphics applications: sketching interfaces and systems used to create 3-dimensional faces, through the mediation of a statistical model. We present our results where we sketch a nose and search for a geometric face model in a database whose nose best matches the sketched nose.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Sketch-based modeling

1. Introduction

Creating 3-dimensional faces is important in a number of areas. This includes reconstructing a suspect in a police investigation from a witness description, creating characters for computer games or films, acquiring topologically consistent faces for research purposes, and creating avatars for gaming or the internet. Using commercial software packages such as Maya requires rigorous training in using intricate tools and operations, which are far from intuitive or straightforward. Additionally there is no guarantee that the faces that are constructed using such packages are anatomically correct.

Sketching is a way to describe a 3-dimensional object on a 2-dimensional surface, whereby the dimension that represents the depth information is not preserved. The art of sketching 2-dimensional depictions of real objects is quite intuitive for most people. Computer-based sketching interfaces aim to do the reverse process of reconstructing 3-dimensional objects using a 2-dimensional sketch. Sketching can therefore provide a high level control to create face models and does not require any modelling knowledge of the user.

We introduce a system that could be used to create 3-dimensional models of faces using 2-dimensional sketching. The facial models produced with this system are anatomically correct and can be loaded into other software packages for further manipulation. We aim to accomplish this by using a knowledge-base of laser scanned faces represented as a statistical model. We will focus primarily on creating faces for the purposes of police investigations although it is also relevant to the other areas mentioned above. Our initial work has focused on the nose. We can sketch models of noses and search for whole faces using the sketched nose. We can also create new faces using a linear combination of known faces to produce a nose that best fits the sketched strokes.

The rest of the paper is organised as follows: Section 2 presents an overview of related work. Section 3 gives an overview of the stages of our approach which are further discussed in Sections 4 and 5. The results of our initial tests are presented in Section 6, Section 7 gives conclusions and plans for further work.

2. Background and Previous Work

2.1. Creating face models

Techniques to create geometric face models have strived to move from low-level methods (e.g. Parke [Par72]) towards more automatic, high-level methods. Today more sophisticated ways are available to transfer a physical description of a person into a digital form. A person's face can be measured automatically using 3D laser scanning although the data often needs cleaning up as it is too detailed and suffers at concave areas and the polar regions. Williams [Wil90] handles missing data by using recursive blur filtering to create a smooth surface where the missing data is replaced with smooth neighbourhood estimates. Ypsilos et al. [YHR04] measures a face automatically by projecting a lighting pattern on the face and recording it using multiple cameras to calculate 3D positions, but similar to laser scanning there are clean-up issues. These methods are quite expensive because they require sophisticated equipment. DeCarlo et al [DMS98] uses statistics based on anthropometric landmarks and measurements to generate faces by fitting a prototype face model using the landmarks. By randomising the measurements, plausible faces can be generated. This can populate an entire world of actors but their solution does not however produce faces with specific features chosen by a user. Similarly Rudomin et al [RBC02] can generate plausible face models but instead of using anthropometric measurements they use an eigenface (Principal Components Analysis) approach based on MPEG-4 feature points that have been marked on a set of 3-dimensional face models. They can generate new individuals by randomising the eigenvalue weights that specify how much each eigenface contributes to a reconstructed face. The weights are restricted to vary within ± 3 standard deviations from the mean to avoid unrealistic faces. Blanz and Vetter [BV99] use Principal Components Analysis (PCA) to construct a face space, where each of the 128 principal components is one standard deviation in magnitude, and by navigating along these axes, new faces can be generated by adding different deviations to the average face located at the origin. They can fit a face model to a single photograph and furthermore map facial attributes to their parameter space to acquire desired properties such as thick face versus thin face, more masculine, more feminine etc. Chen and Fels [CF04] offer a gradient-based navigation through a space of 3-dimensional faces generated using FaceGen [Sin]. A combination of dynamic and static sliders are used to travel through a space of the 3D faces starting from the average face. They base their navigation on Adobe's approach to selecting colours, a colour wheel and sliders. Blanz et al [ABHS06] introduces a system that can be used to create face models from incomplete witness description based on [BV99]. By starting with the average face, the user can select individual features and navigate the gradients of conformation parameters through sliders that affect the selected facial features and characteristics. To assist with creating full face models from incomplete descriptions, a correlation between facial features is learned from the existing models. This way unspecified parts are automatically filled in and every change affects the whole face, unless constraints are enforced, to give the most natural and realistic outcome.

2.2. Faces for police work

A popular approach in witness identification is to construct faces by blending together individual facial features. EFIT and 3D-EFIT [Asp], PROFit [ABM] and Identi-Kit [Ide] are commercial packages widely used by police and security organisations and operate on vast libraries of facial features. Numerous papers report on the fallacy of these systems. Laughery and Fowler [LF80] conducted a study where subjects were shown photographs of individuals and then worked with sketch artists and technicians creating Identi-Kit composites to construct the target face. They found that the artist's sketches were superior to the composites. They also constructed the target faces by viewing the photos directly which gave a better result from the artists but had no effect on the composites. They also maintain, along with [Bad79, Fro02, BM96], that a major flaw with these systems is that they are primarily a feature-based approach. The user selects individual facial features that combined give a representation of a face as opposed to the holistic manner people are believed to store and perceive faces, relying more on shapes and relationships between features, i.e. their spatial configuration. A series of experiments carried out on an individual that suffered from visual object agnesia and dyslexia, but had normal face recognition, show that the crucial information for facial identity is the spatial configuration of the main internal features and the spatial relations to each other [Mos97]. Frowd [Fro02] attempts to overcome some of the issues that plague composite systems with his software EvoFIT. This employs a holistic face model based on Principal Components Analysis with a genetic algorithm to evolve a selection of faces presented to the user that are close to the desired target and free of unrealistic features. It uses control points to morph between features in order to generate different facial shapes. Studies show that it had significant benefits over other composite systems but like the others it works with images and not 3D models.

2.3. Sketching faces

Xiao [Xia04] introduces a method to construct simple objects through sketch recognition. The method relies on predetermined stroke features that specify object components. These metrics are normalised and then classified using K-means clustering. The face consists of 9 strokes in any order corresponding to 9 predetermined components that make up the face model (face, eyes, eyebrows, nose etc). Therefore the face has 9 classes that the gesture recognition must identify when applying the K-means. The classified strokes are then used to construct 3D objects where each shape is rendered as a simple 3D object. Chiu's [Chi05] SketchEx can construct a 3D face with an expression by sketching simple lines that represent different expressions. These lines are interpreted as parameters that are sent to an external module which constructs the closest facial expression on a default head model. Sharon and Panne [SvdP06] train a probabilistic model on examples of sketches based on classes of constellation models. This method assumes that similar parts are drawn with similar strokes where each part is labelled. It uses a maximum likelihood search to try to identify the object class and labels.

3. Our Approach

Our system comprises of two parts: An offline part and an online part. The offline part is where data is collected and processed to form a knowledge-base in the form of a statistical model that can be accessed in real-time by the online part. The online part is an interactive sketching interface that can interact with the generated, statistical model to provide an intelligent feedback to any sketched strokes. The stages of our approach are as follows:

- **Gathering and standardising data** - We have a set of laser scanned face models [BV99], which we will use as our knowledge-base. The laser scanned models have no comparable properties so we label important anthropometric features on the face using feature points (FPs). Furthermore by placing a corresponding set of FPs on a standard head model, we can deform that model to approximate the laser scanned head models giving us a set of head models that share the same vertex topology.
- **Constructing a statistical model** - Using the comparable feature point data produced by the standardisation process, we construct a statistical model using Principal Components Analysis (PCA).
- **Extracting information from sketched strokes** - We analyse sketched strokes on a featureless template head model by locating key locations from the strokes which correspond to the anthropometric feature points. This information is sent to the statistical model for pattern recognition.
- **Generating a head model** - After the closest match in the statistical model has been identified the appropriate standardised head model is fitted to the strokes.

4. Statistical Model

4.1. Standardising data

We have a database of 138 laser scanned face models to generate the information that we will use to train the statistical model. There are several issues with using the face models directly. The meshes are constructed using evenly spaced vertices because of the scanning process. This is not an efficient way of constructing a face mesh and therefore the meshes are far too detailed. That means any process handling and manipulation of the models would be too time consuming and not suitable for a real-time application. More importantly, they do not share a common vertex topology which makes any comparison difficult. It means representing the mesh using a linear combination is not possible, and they do not share any common, measureable characteristics so using it with a statistical model is therefore troublesome. Additionally, the models are often asymmetrical whereby facial features are partly missing on either side, and finally they are not a complete head shape (the laser scans only include the face masks) which is what we are after.

We rectify this by applying a Radial Basis Function (RBF) interpolation, the concept of which is shown in Figure 1. We use a clean mesh of a 3-dimensional head, which we call a standard head (top left), and deform it to approximate the structure of every scanned face model (top right) in the database. This gives us a collection of complete head models that share the same vertex topology. The quality or polygon complexity of the standard head should be good enough to allow constructing any facial features with ease without sacrificing the possibility of real-time calculations and rendering.

This problem can be thought of as a method of interpolation of feature points (FPs) where both the standard face model and the laser scanned models have n corresponding FPs that represent important facial features. Say that $p_i \in R^3$ and $q_i \in R^3$, $i = 1,\ldots,n$ where p_i are the source FPs that lie on the standard face and q_i are the target FPs that lie on the laser scanned face. We need a function f that maps the feature points p_i to q_i, and provides interpolation for intermediate vertices so as to deform the standard face model to approximate the laser scans. The FPs we are going to use are shown as red circles in Figure 1 and chosen in accordance to craniofacial landmarks on the head and face given by Kolar and Salter [KS96]. El-Hussana [EH03] argues that using anthropometric landmarks is an efficient way of modelling human face variations and shows that the eigenface (PCA) variations have a reasonable Gaussian distribution around the mean face. However these feature points are placed manually and the exact location on the model can often be difficult to find [ACC*03], and this affects the overall accuracy and quality of the interpolated face. The RBF Network is of the form

$$f(p_i) = \sum_{j=1}^{n} c_j \theta_j(p_i) + Rp_i + t, \tag{1}$$

where $\theta_j(p_i) = ||x_i - x_j||$, p_i is a feature point i, and $||x_i - x_j||$ is the euclidean distance between feature points i and j where $i \neq j$. x_i and x_j are 3-dimensional coordinates of FPs i and j respectively.

It therefore consists of a weighted linear combination of n basis functions θ_i where the value of the function depends only on the distance from its centre, and an additional polynomial (affine transformation) factor where $R \in R^{3\times3}$ adds rotation, skew and scaling, and $t \in R^3$ is a translation component. This assures a certain degree of polynomial precision to avoid a poor approximation of the unknown function away from the feature points [KHYS02, PHL*98]. $c_j \in R^3$ are the linear weights we are trying to determine by using the feature points which we will then use to interpolate the

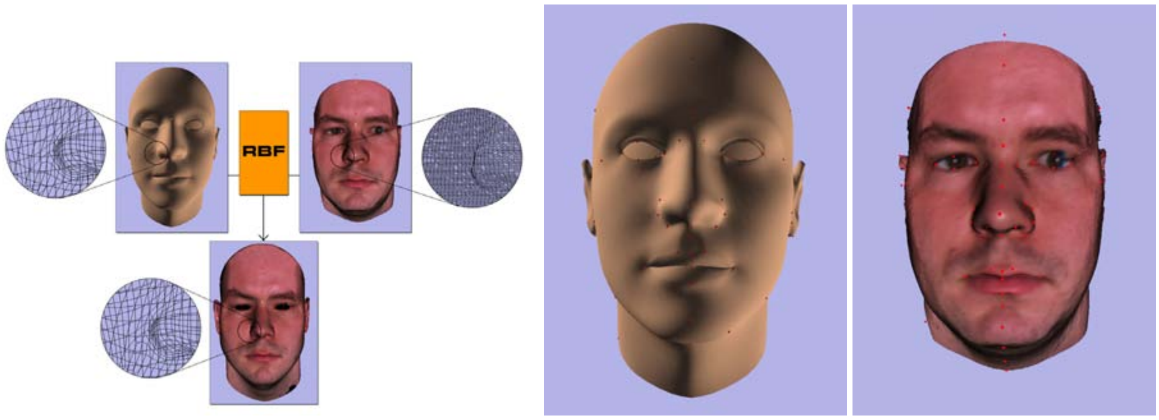

Figure 1: *The RBF interpolation process on the left deforms a standard head model to approximate a laser scanned model. The two images on the right show the landmarked FPs (red circles) on the standard head and laser scan.*

vertices in the model. Zhang [Zha06] compares the result using different basis functions and finds that the multi-quadric gives the best approximation, but we have found as suggested by [LEKM03] that the inverse multi-quadric gives a better result

$$\theta_j(p_i) = (||p_i - p_j||^2 + \delta_j^2)^{-\mu},\ \mu > 0, \tag{2}$$

where δ_j is the stiffness constant measured by the euclidean distance between p_j and the nearest p_i as suggested by Eck [Eck91]

$$\delta_j = min||p_i - p_j||,\ i \neq j. \tag{3}$$

This leads to a smaller deformation on widely scattered feature points and larger deformation for closely scattered ones. Since we use the inverse multi-quadric basis function, $\mu = \frac{1}{2}$. To remove the affine deformation from the radial basis functions we add the following constraints:

$$\sum_{j=1}^{n} c_j = 0 \quad \text{and} \quad \sum_{j=1}^{n} c_j p_j^T = 0.$$

This system can then be solved using a linear solver such as Singular Value Decomposition or Least Squares [GVL96].

4.2. Constructing a statistical model

We use an eigenface approach developed by Turk and Pentland [MT91] to construct a statistical model based on Principal Component Analysis (PCA). They applied this method to detect and identify human faces in images and introduced the concepts eigenface and face-space, where the eigenfaces are in fact the principal components from the training data. We however, will train the model on the FPs that were used to deform the standard head. Each FP consists of a 3-dimensional coordinate specifying its location in euclidean space. For our tests in this paper we will only be using FPs that relate to the nose area. We will still use the terms eigenfaces and face-space despite the difference in the type of data used.

We start by finding the mean face. If $\Gamma_1,..,\Gamma_m$ is the training set of face FPs where m is the number of faces and $\Gamma_i\ \varepsilon\ R^{n\times 3}$, $i = 1,...,m$, then the mean face is defined as $\Psi = \frac{1}{m}\sum_{i=1}^{m}\Gamma_i$. Each face (or FPs) differs from the the mean by the vector $\Phi_i = \Gamma_i - \Psi$. The difference vectors Φ are then used to find a set of m orthonormal vectors u_k and their eigenvalues λ_k that are best fit to represent the distribution of the dataset. This is done by constructing a covariance matrix C and calculating its eigenvalues and eigenvectors using

$$\begin{aligned} C &= \tfrac{1}{m}\textstyle\sum_{i=1}^{m}\Phi_i\Phi_i^T \\ &= AA^T, \end{aligned} \tag{4}$$

where $A = [\Phi_1\ \Phi_2\ \ldots\ \Phi_m]$. The eigenfaces E are found using

$$E_i = \sum_{j=1}^{m} u_{ij}\Phi_j,\ i = 1,...,m. \tag{5}$$

A subset m' of eigenfaces (eigenvectors) are chosen according to significance, i.e. the ones with the highest eigenvalues. The chosen eigenvectors make up a m'-dimensional orthonormal space referred to as face-space. To classify whether or not a new arbitrary set of FPs (Γ) is recognised as a face or not is done by first transforming the data into its eigenface components, what [MT91] calls projecting the face into face-space. It basically means that the system tries to reconstruct the FPs by using a linear combination of the m' chosen eigenfaces found by

$$\omega_k = E_k^T(\Gamma - \Psi),\ k = 1,...,m'. \tag{6}$$

The ω values correspond to the weight that each eigenface k contributes to create the new input face, and $\Omega^T = [\omega_1\omega_2...\omega_{m'}]$. We can then multiply the weight vector Ω

with the eigenface vector E to get the projected face Φ_f. The eigenfaces not used because of low eigenvalue are assigned with a weight 0. By measuring the distance ε between the projected face and the mean adjusted input face Φ, where $\Phi = \Gamma - \Psi$, we can make judgements about the input face. Whether the input face is recognised as a face or not depends on whether or not the distance ε is below a determined threshold.

$$\varepsilon^2 = ||\Phi - \Phi_f||^2. \tag{7}$$

To make a face identification, the weight vector Ω_i, $i = 1,..,m$ for each face data in the training set is calculated beforehand and stored. When the weights Ω for the input face are calculated we calculate the Euclidean distance

$$\varepsilon^2 = ||\Omega - \Omega_i||^2, \ i = 1,..,m. \tag{8}$$

The minimum distance is chosen and if it is below a determined threshold, the face is identified as the face belonging to the weight values Ω_i. New faces can be constructed by adding a linear combination of eigenfaces to the mean face. A new face F_n is

$$F_n = \Psi + \sum_{i=1}^{k} \Omega_i E_i, \tag{9}$$

where k is the number of eigenfaces used in the reconstruction.

5. Sketching Interface

5.1. Sketching strokes

We have developed a simple 3-dimensional interface where a user can sketch 2-dimensional strokes on a 3-dimensional template object (see Figure 2 using either a mouse or a pen input device. We distinguish between a 2-dimensional screen coordinate space (x,y) where the sketching takes place, and a 3-dimensional object space (x,y,z) where the template object lies.

When the user is sketching in screen space, the cursor or stylus movements are tracked by creating 2-dimensional control points that will make up the stroke. The stroke begins when the pen touches the surface or the mouse button is clicked until the pen is lifted or the mouse button released. Upon completion, the stroke is projected into the 3-dimensional object space and stored as a 3-dimensional stroke, again using control points but now with the added third dimension containing the depth. In our current experimentation, strokes where the endpoints do not touch a part of the template object surface are irrelevant and therefore ignored. This is because the aim here is to draw facial features on the template object. The constraint is verified by projecting the (x,y) coordinates into the 3-dimensional space and checking if intersection with the template object occurs. If both endpoints intersect then the stroke is valid and the (x,y,z) object space coordinates are calculated and stored as 3-dimensional control points,

$$C_i = \left[\begin{array}{cccc} p_1 & p_2 & \dots & p_n \end{array} \right] \ \varepsilon \ R^{n\times 3}, \tag{10}$$

where p_k is a 3-dimensional coordinate for a single control point. Additionally, $i = 1..m$ where m is the number of sketched strokes.

A valid stroke is then projected into the 3-dimensional object space. If all control points intersect with the template object the calculation is straightforward using the projection. It is effectively shooting a ray from the current camera viewpoint through all the control points and return the (x,y,z) coordinates where the ray intersects a polygon on the template object. A problem arises when a ray going through a control point does not intersect any polygons in the object space. The constraint forcing the endpoints on any stroke to touch a polygon on the template helps to overcome this issue. The endpoints lie on the surface and have a known depth so the coordinates for the remaining control points are interpolated using a cubic Catmull-Rom spline interpolation [SP95] defined as

$$f(t) = [1 \ t \ t^2 \ t^3] \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ -0.5 & 0.0 & 0.5 & 0.0 \\ 1.0 & -2.5 & 2.0 & -0.5 \\ -0.5 & 1.5 & -1.5 & 0.5 \end{bmatrix} \begin{bmatrix} P(0) \\ P(1) \\ P(2) \\ P(3) \end{bmatrix},$$

where $P(0),..,P(3)$ are the control points on a Catmull-Rom spline and $t = 0.0$ is defined at $P(1)$ and $t = 1.0$ at $P(2)$. We specify the first control point to be $P(0)$ and $P(1)$, the last one as $P(2)$ and $P(3)$, and then the depth values for the intermediate points are calculated by assigning t as the number of the current point divided by the total number of intermediate points.

5.2. Extracting information from sketched strokes

In our tests the user is presented with a template model which represents a featureless head in the sketching interface as described earlier. The user then sketches 5 strokes that are usually suggested by artists as the basic outlines when drawing a simple nose and are shown in Figure 2. To simplify the process for the sake of testing, the strokes are to be sketched in a specific order indicated by the numbers in the Figure. The strokes are projected to a 2-dimensional plane, and analysed to find feature points (FPs) that correspond to the anthropometric landmarks used to train the statistical model as described in the previous section. Figure 2 shows the detected feature points and their numbering. The feature points are found by using a simple heuristic search.

After the FPs have been located, normalised distances between them are calculated and analysed using the statistical model. The statistical model calculates the eigenvalue weights which are used to make judgements about the sketched strokes. If the distance ε is within a certain threshold then the strokes are recognised as representing a nose

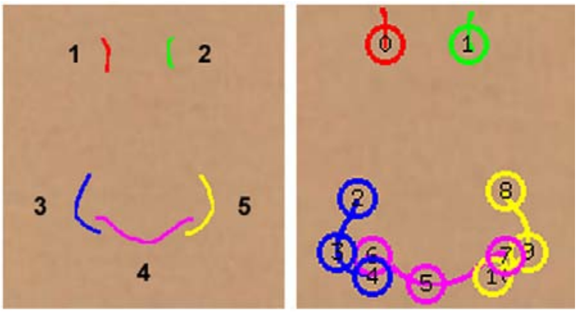

Figure 2: *Required strokes and their number in the sequence (left). Heuristically located FPs (right)*

shape. Then, using the calculated weights, the minimum distance to the pre-calculated coefficients (Ω_i) is found to identify which of the noses in the training set is closest to the sketched nose.

6. Experiments and Results

We devised two simple experiments which we will refer to as A and B. In experiment A, we sketch the nose outlines as described in the previous section on a featureless template head and pull out the person in our training set whose nose best matches the sketch. It operates in 3D as the eventual aim is to allow sketching from any angle. In experiment B, we map a texture of individuals in the training set onto a 2-dimensional plane as opposed to the 3-dimensional environment in experiment A. This simplifies the process. We trace the nose outline to see if the recognition process accurately identifies the correct individual when a sketch faithfully represents the shape of the nose. In section 6.3 we show how we can create new 3D faces that can give a better fit to the 3D sketch.

6.1. Experiment A

Figure 3 shows examples of results from experiment A. The user sketches a nose on a template head (left) and the system finds the closest match in the training set and fits the corresponding model (right) to the sketched strokes. Generally, this finds a model with a nose that has similar characteristics as the sketch. However, it is limited to the number of noses that exist in our knowledge-base which means it is not a perfect match.

6.2. Experiment B

When the system pulls out a face after the user sketches the nose outlines, we need to determine if it really pulls out a face that has the outlines that accurately describe the nose. We can test this by tracing the nose outlines on the texture maps that correspond to the individuals in the training

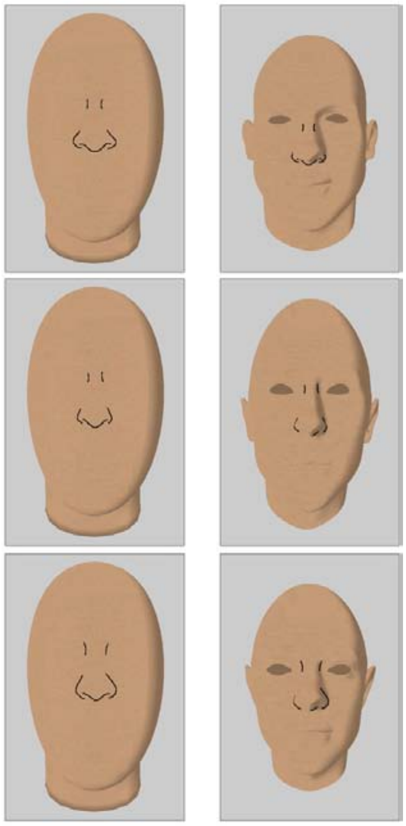

Figure 3: *Some results from experiment A. The user draws a nose on the template head on the left, and the system finds the model in the training set whose nose is the best match.*

data. This guarantees that the sketch faithfully represents the nose we are after. After tracing the nose, the system tells us which face in the statistical model was recognised from that sketch and that way we can see if the best match was indeed the individual on the texture map. This way we can test whether or not there is direct correspondence between the 2-dimensional sketch and the data in the statistical model. Figure 3 shows examples of the results from experiment B. The number of the individual is given below each person and the top five matches from the statistical model are displayed to the right ranked from top to bottom where the top number is the best match. Also the number of the person being traced is higlighted in red to show where it landed on the list. In most cases, the correct individual was found. Using a tablet and pen instead of a mouse gave considerably better chance of recognition as the sketches were more accurate. Lazy and sloppy sketching immediately resulted in a poor recognition rate. A subtle deviation when tracing the outline can result in a failed recognition. This is shown in Figure 4 where the second attempt to trace subject 3518 fails when the second best match from before is now considered the best match. This

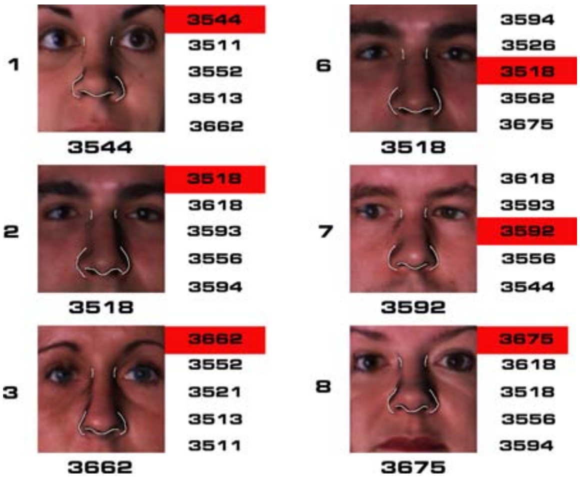

Figure 4: *Some results from experiment B. The user traces a nose from an image of an individual in the training set (ID shown below image), and the system finds the top five matches (ranked from top to bottom).*

sensitivity implies that more parameters are needed than the feature points can provide, as they only represent a discrete coordinate in space.

6.3. Creating a new face

We can create new faces using a linear combination of eigenfaces according to Equation 9. Here, $\Omega_i, i = 1, .., m$, are the eigenvalues or linear weights, acquired when projecting the interpreted sketch onto the space of known faces using Equation 6, while the mean face and eigenfaces belong to a different eigenmodel created from the complete set of FPs. A similar approach using photographs and sketches is used by Tang and Wang [TW04]. F_n is then subject to RBF interpolation which produces a model that gives a better fit to the sketched strokes than any single model extracted from the data set could give. Figure 5 shows examples of 3 faces created using this procedure.

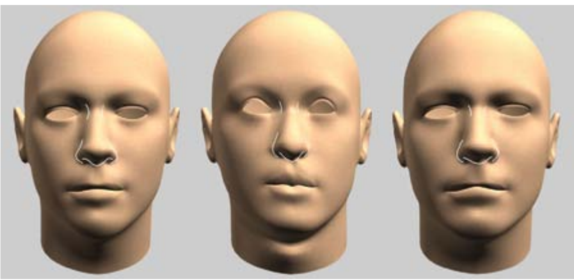

Figure 5: *3 faces created using a linear combination (Eq. 9)*

7. Conclusions and Future Work

We have presented a new approach to creating 3-dimensional face models through 2-dimensional sketching. Initial work on noses enables the creation of a face model which had a nose whose structure resembled the sketched nose. This approach could be extended to the whole face where the shape of the face model and every facial feature is created through sketching.

A number of improvements need to be made. We only experimented with the nose and simplified the experiments by constraining the user to draw specific strokes in a correct sequence. Instead the user should be free to sketch any combination of facial features using an arbitrary number of strokes in any order to produce the 3D model he is after. In our experiments, we simply pulled out the individual from the training set that most closely resembled the sketched input. This puts a limit on the number of features that can be created. Therefore we experimented with producing gradient levels by using a linear combination of models in the database which creates a model that fits the sketched strokes better.

Currently there is an issue when it comes to create outlier faces, i.e. faces that deviate far from the mean face. This could be fixed by improving the structural resemblance between the two eigenmodels used. The FPs are limited in the sense of how descriptive they are with regards to the facial features and their shape. Therefore using the FPs in the statistical model puts limits on the possible sketching manoeuvres and increases sensitivity when distinguishing between the sketched facial features. We are currently investigating using curvature rather than FPs as the input data for sketching.

The inital tests used 50 heads as our training set and the facial extraction and construction operates in real-time. We intend to extend the training set to operate on a database of 2000 heads we are currently processing. This may lead to computational issues as a result of a larger data set.

References

[ABHS06] ALBRECHT I., BLANZ V., HABER J., SEIDEL: Creating Face Models from Vague Mental Images. *Eurographics 2006 25*, 3 (2006).

[ABM] Profit. http://www.abm-uk.com. ABM.

[ACC*03] AISHA F., CANT E., CHAKRAVARTHY A., DHANALAKOTA R., HAN D., WU C.: Feature Point Placement on MPEG-4 Compliant Face Meshes. *Darwin Project, Dep. of Computer Science, Univ. Sheffield* (2003).

[Asp] Efit. http://www.efit.co.uk. Aspley Identifying Solutions.

[Bad79] BADDELEY A.: The limitations of human memory: implications for the design of retrospective surveys.

The Recall Method in Social Surveys. London: University of London Institute of Education (1979), 13–27.

[BM96] BRUNELLI R., MICH O.: Spotit! an interactive identikit system. *Graphical Models and Image Processing 58*, 5 (1996), 399–404.

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 187–194.

[CF04] CHEN T.-P. G., FELS S.: Exploring gradient-based face navigation interfaces. In *GI '04: Proceedings of the 2004 conference on Graphics interface* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004), Canadian Human-Computer Communications Society, pp. 65–72.

[Chi05] CHIU J.: *SketchEx: Sketch-based Interface for 3D Face Modeling*. Master's thesis, University of California at Los Angeles, 2005.

[DMS98] DECARLO D., METAXAS D., STONE M.: An anthropometric face model using variational techniques. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 67–74.

[Eck91] ECK M.: Interpolation methods for reconstruction of 3d surfaces from sequences of planar slices. *CAD und Computergraphik 13(5)* (1991), 109–120.

[EH03] EL-HUSSUNA A.: *Statistical variation of Three Dimensional face models*. Master's thesis, IT-University of Copenhagen, 2003.

[Fro02] FROWD C.: *EvoFIT: A Holistic, Evolutionary Facial Imaging System*. PhD thesis, Univ. of Stirling, 2002.

[GVL96] GOLUB G. H., VAN LOAN C. F.: *Matrix Computations, 3rd edition*. The Johns Hopkins University Press, 1996.

[Ide] Identi-kit. http://www.identikit.net. Identi-Kit Solutions.

[KHYS02] KAHLER K., HABER J., YAMAUCHI H., SEIDEL H.-P.: Head shop: Generating animated head models with anatomical structure. *SIGGRAPH* (2002), 55–64.

[KS96] KOLAR J., SALTER E.: Craniofacial Anthropometry: Practical measurements of the head and face for clinical, surgical and research use. Charles C, 1996.

[LEKM03] LORENZO M. S., EDGE J. D., KING S. A., MADDOCK S.: Use and re-use of facial motion capture data. *Vision, Video and Graphics* (2003), 1–8.

[LF80] LAUGHERY K., FOWLER R.: Sketch artist and Identikit procedures for recalling faces. *Journal of Applied Psychology 65*, 3 (1980), 307–316.

[Mos97] MOSCOVITCH M.: What Is Special about Face Recognition?: Nineteen Experiments on a Person with Visual Object Agnosia and Dyslexia but Normal Face Recognition. *Journal of Cognitive Neuroscience 9*, 5 (1997), 555–604.

[MT91] M. TURK A. P.: Face recognition using eigenfaces. In *IEEE Computer Vision and Pattern Recognition* (1991), pp. 586–590.

[Par72] PARKE F. I.: Computer generated animation of faces. In *ACM'72: Proceedings of the ACM annual conference* (New York, NY, USA, 1972), ACM Press, pp. 451–457.

[PHL*98] PIGHIN F., HECKER J., LISCHINSKI D., SZELISKI R., SALESIN D. H.: Synthesizing realistic facial expressions from photographs. *Computer Graphics 32*, Annual Conference Series (1998), 75–84.

[RBC02] RUDOMIN I., BOJÓRQUEZ A., CUEVAS H.: Statistical generation of 3d facial animable models. *Proceedings of the Shape Modeling International 2002 (SMI'02)* (2002).

[Sin] Facegen. http://www.facegen.com. Singular Inversion inc.

[SP95] SHIKIN E. V., PLIS A. I.: *Handbook on Splines for the User*. CRC Press Inc, 1995.

[SvdP06] SHARON D., VAN DE PANNE M.: Constellation models for sketch recognition. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2006), pp. 19–26.

[TW04] TANG X., WANG X.: Face sketch recognition. *Circuits and Systems for Video Technology, IEEE Transactions on 14*, 1 (2004), 50–57.

[Wil90] WILLIAMS L.: Performance-driven facial animation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1990), ACM Press, pp. 235–242.

[Xia04] XIAO D.: *Sketch-based Instancing of Parameterized 3D Models*. PhD thesis, The University of British Columbia, 2004.

[YHR04] YPSILOS I. A., HILTON A., ROWE S.: Video-rate capture of dynamic face shape and appearance. In *FGR* (2004), pp. 117–122.

[Zha06] ZHANG Y.: An efficient texture generation technique for human head cloning and morphing. In *GRAPP 2006, First International Conference on Computer Graphics Theory and Applications* (2006), Insticc Press, pp. 267–275.