EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2008)
C. Alvarado and M.- P. Cani (Editors)

Sketching Faces

Orn Gunnarsson and Steve Maddock

The University of Sheffield, England

Abstract

Faces can be modelled using a number of techniques. Existing faces can be transferred to a digital form using equipment such as laser scanners. New faces can be constructed using commercial modelling tools, or using specialist software, e.g. Photo-fit software. We present a technique that can create a 3D head using intuitive 2D sketching techniques. This involves bringing together two types of graphics applications: sketching interfaces and systems used to create 3D faces, through the mediation of a statistical model.

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

1. Introduction

Creating 3D faces is important in a number of areas. This includes creating characters for computer games or films, acquiring topologically consistent faces for research purposes, creating avatars for gaming or the internet, and reconstructing a suspect from a witness description in a police investigation. To construct faces 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 3D object on a 2D surface, whereby the dimension that represents the depth information is not preserved. Computer-based sketching interfaces aim to do the reverse process of reconstructing 3D objects using a 2D 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 an approach that can be used to create 3D models of faces using 2D sketching on 3D surfaces. The face models produced with this system are anatomically correct and can be loaded into other software packages for further manipulation. We accomplish this by using a knowledge-base of laser scanned faces represented as a statistical model. This method improves on our earlier work [GM07] by using a more advanced statistical model and input method.

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. A demonstration of the system being used to manipulate a face model is shown 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, a person's face can be measured automatically using 3D laser scanning, or by projecting a lighting pattern onto the face and recording it using multiple cameras to calculate 3D positions [YHR04]. Both, however, require subsequent data clean up measures, and rely on expensive equipment.

DeCarlo et al [DMS98] randomises statistics based on anthropometric landmarks and measurements to generate plausible faces by fitting a prototype face model using the landmarks, although no specific facial features can be 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 (PCA)) approach based on MPEG-4 feature points that have been marked on a set of 3D face models. They can generate new individuals by randomising the eigenvalue weights that specify how much each eigenface contributes to a reconstructed face. Blanz and Vetter [BV99] use 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 slider-controlled, gradient-based navigation through a space of 3D faces.

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 [Vis], and FACES [IQ] 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 [Bad79,Fro02]. A major flaw with these systems is that they are primarily a feature-based approach. Instead people are believed to store and perceive faces in a holistic manner, relying more on shapes and relationships between features, i.e. their spatial configuration [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.

2.3. Sketching faces

Sketching interfaces for modelling typically fall into one of two categories, one where no prior knowledge is assumed and one or more strokes are interpreted based on arithmetic functions that make up or edit an object. Free-form systems like Teddy [IMT99], Plushie [MI07] as well as mesh editing systems like [NSACO05, ZNA07], fall into this category. The latter category assumes prior knowledge about what is possible to produce within the system framework. This is often referred to as templates. Using templates generally puts more restrictions on the sketching process as it is limited to a certain class of objects, but makes it possible to create more intricate objects of the particular class. Yang et al [YSvdP05] is an example of a simple, parameterised template system which classifies an object by matching nodes and edges with template classes. Kokai et al [KFS*07] uses a template based approach to create automotive vehicles using existing example shapes. Company et al [PCN05] and Igarashi et al [LDI06, Iga07] discuss a wide range of sketching interfaces and how they operate.

We argue that a template approach is ideal for sketching faces as they are complex but all contain the same set of characteristics. Few sketching interfaces involve sketching and creating faces, and none of them offers a flexible way of creating accurate 3D 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. A 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 and it uses a maximum likelihood search to identify the object class and labels. Sketching is useful to animate faces where sketched strokes indicate changes to facial expression that make up a series of keyframes. Chang and Jenkins [Cha06] use reference and target curves to specify mesh deformation. It offers flexibility and requires no prior knowledge. Using prior knowledge can produce more natural results but limits the scope to lie within the range of the acquired database. FacePoser [LCXS07] uses a knowledge base made up of pre-recorded facial expressions in a maxmimum likelihood framework to map sketched strokes to the most probable and natural outcome given their database of expressions.

3. Our approach

Our approach is made up of an offline part and an online part. The offline part (Section 4) is where face 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 statistical model to provide intelligent feedback to any sketched strokes (Section 5). The stages of our approach are as follows:

- **Gather and standardise face data** - We use a set of laser scanned face models [BV99] as our knowledge-base. Feature points (FPs) and Radial Basis Functions (RBFs) are used to deform a standard head model to approximate the laser scanned head models, giving us a set of models that share the same vertex topology (see Section 4.1).
- **Construct a statistical model** - Using the set of feature point data produced by the standardisation process, we train two mixtures of probabilistic principal component analysers on the FP coordinates (see Section 4.2).
- **Interpret sketched strokes from a user** - The user sketches on a 3D head model starting with a template head, although any head in the database range could be used as a starting point. We project and analyse sketched strokes by finding and collecting key locations that correspond to the set of FPs (see Section 4.3).
- **Find best fit and update model** - Using the statistical model, information is collected from the sketched strokes is validated and used to generate the most plausible 3D facial features given the input data (see Section 4.4). These features are represented using the FPs and the head model is updated using these new features through RBFs. The user can now continue sketching.

The behaviour and flexibility of the system is determined by what data is extracted and used from the set of laser scanned face models. For the purpose of this paper, we use the FPs that were used to standardise the models. Since the models have been labelled with a set of FPs, their internal structure could be deformed by picking and moving the FPs where they are treated as RBFs (similar to [BK05]). Instead of moving FPs around, sketching can be used to indicate the desired position of more than one FP in one stroke.

We want to make sure that only plausible faces can be generated so free-form manipulation of the FPs is not permitted. The model has to be constrained to only allow fairly realistic faces. We approach this by using a Gaussian mixture model that can determine whether a generated face is probable or not. It also plays an important role in the presence of incomplete input data and provides correlation between facial features.

A key element is how to identify and collect data from a set of sketched strokes that can be used to find a likely match in the statistical model. In many sketching interfaces, heuristic methods are used to find key locations, for instance by examining the curvature and timing of strokes. We use a statistical approach where each point in a sketched stroke is subject to maximum-likelihood, as explained in Section 4.3.

Another thing to consider is how a face or facial feature is classified given a set of strokes. A straightforward approach is to find the closest match in our database whereby the system acts as a searching tool. Another possibility is to find the closest match for each feature and blend them together but this raises an issue concerning the labelling of distinct features. The features generally correlate slightly with each other so this poses a risk of creating odd boundaries between features if they are vastly different. A more suitable approach is to determine if the sketched strokes are probable and fill in the rest with the most likely values. This method is better at dealing with incomplete data and gives a more realistic correlation and boundaries between features, as described in Section 4.4.

4. Statistical model

4.1. Standardising data

We use a clean mesh of a 3D head from [Sin], which we call a standard head, and deform it to approximate the structure of every scanned face model in the database. 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. These target FPs form the data set that we feed into the statistical model.

We need a function f that maps the feature points p_i to q_i, and provides interpolation for intermediate vertices. The FPs we use are shown as green circles in Figure 1, and are 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, and this affects the overall accuracy and quality of the interpolated face. The RBF Network is of the form

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

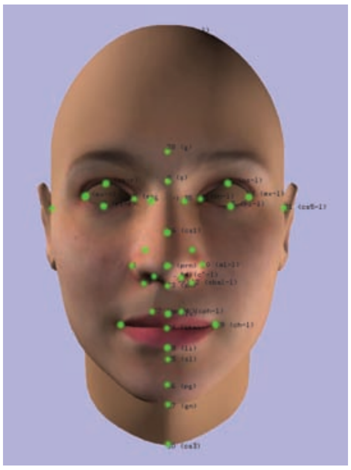

Figure 1: *Landmarked FPs (shown in green).*

where $\theta_j(p_i) = ||p_i - p_j||$, p_i is a feature point i, and $||p_i - p_j||$ is the euclidean distance between feature points i and j where $i \neq j$.

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 then use to interpolate the vertices in the model. Zhang [Zha06] compares the result using different basis functions and finds that the multiquadric 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. 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:

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

4.2. Gaussian mixture model

Principal Components Analysis (PCA) is a popular approach in computer vision where high dimensional data is decorrelated and approximated using a lower dimensional space where each dimension is orthogonal to each other to maximise variance. However, conventional PCA suffers from many limitations. Importantly it is not a density model so it cannot be used with bayesian inference, it uses euclidean distance for classification, it cannot handle missing data, and it cannot be extended to a mixture model which can be used to estimate non-linear projections.

PCA can be defined in a maximum-likelihood framework based on a Gaussian latent variable model to derive a Probabilistic PCA (PPCA) [TB99]. A latent variable model linearly maps an observed d-dimensional vector $\mathbf{t}$ to a q-dimensional, Gaussian latent variable $\mathbf{x}$, d > q, such that

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \mu + \varepsilon \tag{4}$$

where ε is a Gaussian, independent noise model $\varepsilon \sim N(0, \psi)$. This means that the observed vectors $\mathbf{t}$ are also Gaussian distributed $\mathbf{t} \sim N(\mu, \mathbf{C})$. By using an isotropic noise model and setting $\psi = \sigma^2\mathbf{I}$, and therefore the model covariance to $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{W}\mathbf{W}^\mathbf{T}$, the columns of $\mathbf{W}$ span the principal subspace of $\mathbf{t}$ after fitting the model. Fitting the latent variable model is typically achieved by marginalising the latent variables and optimising the parameters through maximum likelihood.

Extending this to a mixture model is done by expressing the probability density function as a linear combination of basis functions

$$p(\mathbf{t}) = \sum_{i=1}^{M} \pi_i p(\mathbf{t}|i), \tag{5}$$

where $p(\mathbf{t}|i)$ is a single PPCA model and π_i is the mixing coefficient or prior probability for component i, $\pi_i \geq 0$ and $\sum \pi_i = 1$. $\mathbf{t}$ is the observed input vector, M is the number of clusters or centres and $p(\mathbf{t}|i)$ is the cluster density function. We can find the posterior probability using Bayes Theorem which enables us to determine what cluster a given input stroke belongs to.

The parameters for this mixture model can be determined from the observed FPs through maximum likelihood. This can also be thought of as minimising the negative log-likelihood of the data. This cannot be determined analytically so we use the iterative Expectation Maximisation algorithm. Poorly initialised parameters can result in a local maxima problem so we use the k-medoids method to perform initial clustering [Bis07]. The initial parameters for each centre are then calculated using a subset given by the hard clustering.

4.3. Analysing strokes

Strokes sketched by the user indicate what the facial features should look like, but the mixture model needs relevant key information from the strokes in order to match the strokes with the correct feature. We use the FPs as the training set for a mixture model responsible for classfying and reconstructing facial features so we need to find points in the sketched strokes that correspond to the FPs. Instead of using heuristic methods to find points of interest, we train a mixture model on the whole bundle of FPs where the number of cluster centres are equal to the number of FPs used per face. To guarantee a correct convergence, the centres are initialised as the mean of each FP. This enables us to treat the problem probabilistically. Points with likelihood above a certain threshold are assigned to the cluster with the highest posterior probability. If more than one belongs to the same cluster, the point with the highest probability is used. The assigned cluster then tells us which FP the corresponding point maps to. Figure 2 gives an example of this process where boundaries for an eye are drawn (blue points). For each point the marginal likelihood is calculated and the posterior probability of each cluster is found. Here four points are identified and assigned to their corresponding FP cluster. The assigned clusters are shown in red and the others in green.

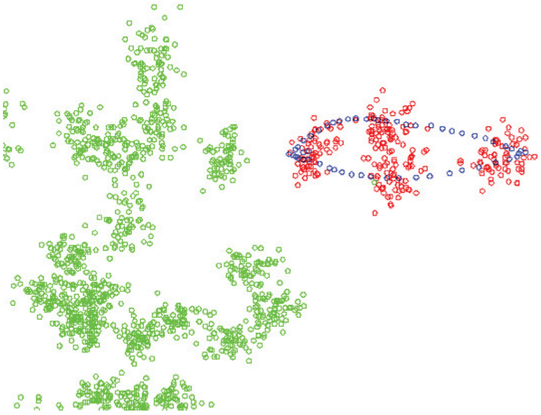

Figure 2: *Sketched points (blue) are mapped to FP clusters through maximum likelihood where the assigned clusters are shown in red and the others in green.*

4.4. Classifying and reconstructing facial features

Upon receiving key points harvested from sketched strokes using the method described in the above section, we use the points to find the best feature to match the strokes. As we explained in Section 3 we are not identifying individuals based on sketching. We want a correlated combination of features to make up a new face. So instead of using the points to find the nearest match in the database of features, we use the observed points as conditional data and marginalise over the missing points to construct a complete set of feature points.

We know from the previous section that the observed points are guaranteed to be probable so the expected data should lie within a range of realistic values. However the threshold can let a combination of points through that might produce some expected points that are less sensible. As a safety, the probability of every expected value is checked, and if it falls below the threshold it is combined with the original coordinates where the mixing coefficient varies depending on the likelihood. The complete set of FPs is then made up of observed, conditional points and expected points. This set of FPs is sent back to the sketching interface which uses it to deform the face to reconstruct the desired facial features. The nature of the FPs returned depends on the number of clusters in the mixture model and is described in the next section.

5. Sketching interface

We have developed a simple 3D interface where at start-up the user is presented with an average 3D head model as a template or alternatively any arbitrary head in the database range. The user can sketch 2D strokes that will project onto the 3D model and be stored as control points. Since the control points lie on a 3D surface the depth ambiguity is partially solved and, since the purpose of any stroke is to represent a feature on the surface, we minimise it further by insisting that one endpoint of a stroke must touch the surface. Because of this the depth value is known where the stroke intersects the model and the remaining depth coordinates are found using a cubic Catmull-Rom spline interpolation [SP95]. There still remain some ambiguities in the sketching process, particularly when sketching from viewing angles that lie between the front view and profile. We allow the user to force the stroke to be projected onto an axis plane which makes it easier to draw profile strokes that could otherwise be projected on a number of different parts on the face.

The basic process of producing a feature from one or more strokes is:

- Strokes are sent to the statistical model which consists of two parts: Stroke analysis 4.3, and feature construction 4.4.
- Local or global influence is preset depending on the number of mixture centres.
- Statistical model returns the best fit as a complete set of FPs.
- Model is updated using the FPs and RBFs where the source model is always the same to avoid incremental changes, and then the user can continue sketching.

The reconstruction and hence the overall feel of the system depends on how many centres the mixture model has been trained on. When the model has few centres, the feature construction is more local and surrounding features are less affected. When there are a large number of centres, the effect is more global with higher correlation between features. By training the model on different preset numbers of centres, a control parameter could be offered allowing the user to vary the degree of localisation. If the user draws large lips which are normally associated with females, any unsketched features become more feminine.

An already sketched feature will not be affected by the global correlation. Therefore, any sketched stroke acts as a constraint while the rest of the unsketched face changes in correlation with the already sketched features. This can be useful in a crime situation where a witness only sees a part of the suspect's face. The hidden features are replaced with the most plausible ones.

Figure 3 shows the difference of sketching a similar, simple stroke using a different number of centres. The picture on the left shows the effect of using few centres where the changes are mainly localised around the stroke, even causing asymmetry although some correlation still takes place. The picture on the right shows the effect of using a larger number of centres where every unsketched feature changes to match the maximum likelihood correlation.

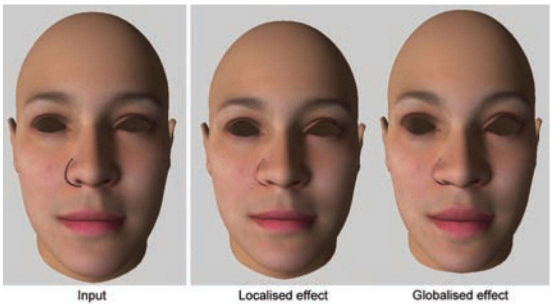

Figure 3: *The effect of using different number of cluster centres in the mixture model. Left: 3 centres results in a more localised effect. Right: 60 centres causes a more global correlation effect where every unsketched feature changes accordingly.*

Two issues that need to be addressed in the sketching process are oversketching and extreme sketching. Oversketching refers to the situation when the user resketches in the same area as a previous sketch, essentially to refine an earlier sketching stroke. We need to decide how to interpret multiple sketch strokes for the same feature. Currently the points in the set of strokes with the highest likelihood are assigned to the suitable FP cluster. A more recent stroke could be given higher priority and the interface could offer the possibility to delete strokes or parts of them but we do not do this at present.

Extreme sketching is when a user sketches a feature in a way that lies outside its natural bounds. Thus, the marginal likelihood drops to zero, which can cause unwanted side effects. Figure 4 shows an example where the middle part of the stroke that indicates where the lowest point of the eye should be is outside the likelihood area. Only points on the stroke indicating the width of the eye are detected and used. However, the user clearly indicated that the lowest point should be changed as well. It could be argued that the information given by the user is not plausible but it might be the case that the points lie only a small distance away from the area of positive likelihood and should not be ignored. We check every stroke to see if it can be adapted to include its unlikely parts, given its likely parts. This makes the sketching process less sensitive to subtle changes and enables extreme sketching as shown in Figure 4.

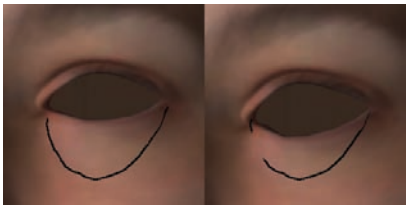

Figure 4: *Here the user intentionally draws an extreme stroke to indicate that he wants the lowest part of the eye to move as far down as possible. (The deformed face hides the sketch stroke.)*

6. Creating faces

Figure 5 shows how small and simple strokes can be used to change facial features. The lip is widened by using a small scribble to indicate its new boundary. After submitting the strokes for processing (to our unoptimised software), it takes from 3-12 seconds to adapt and classify the strokes, and to interpolate and deform the model. The time increases as more strokes are added as any prior process is never cached or optimised. Figure 6 shows how a profile stroke can be used to change the shape of the nose. The strokes act as a guide to the intended shape and the system finds the best fit given the database range. Any number of strokes and features, in any order, can be sketched and processed simultaneously. The strokes can either serve to tweak an existing feature like the ones in Figures 5 and 6, or they can be used to indicate nearly a complete feature as Figure 7 shows. A number of features can be drawn without being processed separately giving the same result as drawing them one by one. Each stroke acts as a conditional dimension in the statistical model when looking for the maximum likelihood solution.

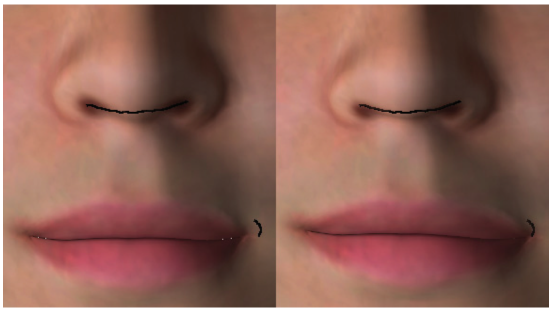

Figure 5: *Using simple strokes to indicate subtle changes to parts of features.*

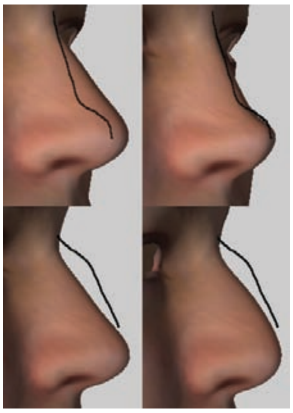

Figure 6: *A new profile silhouette can be indicated with a stroke which is used as a guide to change the shape of the nose.*

7. Conclusions and future work

We have presented a new approach to creating 3D face models through sketching. Sketching acts as a high-level control to modelling where a whole feature can be created by indicating the desired outcome as opposed to going through a series of low level steps. We accomplish this using a template knowledge-base in the form of a statistical model that through a maximum likelihood approach knows how faces are constructed, and can model facial features given minimal input while making sure the outcome is always a plausible face.

However, our current approach is limited to create features where the models have been landmarked with a predetermined set of feature points. The demonstrations in this paper used 39 feature points to describe the main facial features. Facial details cannot be captured using feature points alone and since the points are discrete in space there is a lot of redundant data in the sketched strokes that could be used to enhance classification accuracy so other methods are needed. We are currently experimenting with curvature data with promising results. Using curvature and depth maps makes it possible to create both key features through sketching as well as subtle details anywhere on the face which makes it possible to create a better likeness from a mental image of a face.

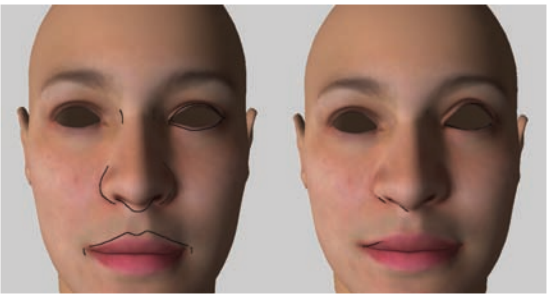

Figure 7: *Our approach allows the user to sketch any number of features at the same time. The most likely features are subject to a set of constraints laid out by the sketched strokes.*

References

[ABHS06] ALBRECHT I., BLANZ V., HABER J., SEIDEL: Creating Face Models from Vague Mental Images. *Eurographics 2006 25*, 3 (2006), 645–654.

[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.

[Bis07] BISHOP C. M.: *Pattern Recognition and Machine Learning*. Springer, 2007.

[BK05] BOTSCH M., KOBBELT L.: Real-time shape editing using radial basis functions. *Eurographics 24*, 3 (2005), 611–621.

[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* (1999), 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* (2004), pp. 65–72.

[Cha06] CHANG E.: *Sketching Articulation and Pose for Facial Meshes*. Master's thesis, Brown University, 2006.

[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* (1998), 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.

[GM07] GUNNARSSON, MADDOCK: A statistically-assisted sketch-based interface for creating arbitrary 3-dimensional faces. In *Proc. Theory and Practice of Computer Graphics 2007* (2007), pp. 123–130.

[Iga07] IGARASHI T.: Sketch-based interfaces for interactive computer graphics. In *SIGGRAPH '07: Course 3* (2007).

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99* (1999), pp. 409–416.

[IQ] Faces. http://www.iqbiometrix.com. IQ Biometrix, Inc.

[KFS*07] KOKAI I., FINGER J., SMITH R., PAWLICKI R., VETTER T.: Example-based conceptual styling framework for automotive shapes. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2007).

[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. Thomas Publisher LTD, 1996.

[LCXS07] LAU M., CHAI J.-X., XU Y.-Q., SHUM H.-Y.: Face poser: Interactive modeling of 3d facial expressions using model priors. In *2007 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (August 2007), pp. 161–170.

[LDI06] LAVIOLA J., DAVIS R., IGARASHI T.: An introduction to sketch-based interfaces. In *SIGGRAPH '06: Course 18* (2006).

[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.

[MI07] MORI Y., IGARASHI T.: Plushie: An Interactive Design System for Plush Toys. *ACM Transactions on Graphics 26*, 3 (2007).

[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.

[NSACO05] NEALEN A., SORKINE O., ALEXA M., COHEN-OR D.: A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph. 24*, 3 (2005), 1142–1147.

[Par72] PARKE F. I.: Computer generated animation of faces. In *Proceedings of the ACM annual conference - Volume 1* (1972), pp. 451–457.

[PCN05] PEDRO COMPANY ANA PIQUER M. C., NAYA F.: A survey on geometrical reconstruction as a core technology to sketch-based modeling. *Computer and Graphics 29* (2005), 892–904.

[PHL*98] PIGHIN F., HECKER J., LISCHINSKI D., SZELISKI R., SALESIN D. H.: Synthesizing realistic facial expressions from photographs. *SIGGRAPH '98* (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), 219–226.

[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.

[TB99] TIPPING M. E., BISHOP C. M.: Mixtures of probabilistic principal component analysers. *Neural Computation 11*, 2 (1999), 443–482.

[Vis] Efit-v. http://www.visionmetric.com. VisionMetric.

[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.

[YSvdP05] YANG C., SHARON D., VAN DE PANNE M.: Sketch-based modeling of parameterized objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2005), pp. 63–72.

[Zha06] ZHANG Y.: An efficient texture generation technique for human head cloning and morphing. In *GRAPP 2006* (2006), pp. 267–275.

[ZNA07] ZIMMERMANN J., NEALEN A., ALEXA M.: Silsketch: Automated sketch-based editing of surface meshes. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2007).