Sketch-Based Posing of 3D Faces for Facial Animation

Orn Gunnarsson and Steve Maddock

The University of Sheffield, England

Abstract

This paper presents a novel approach to creating 3D facial animation using a sketch-based interface where the animation is generated by interpolating a sequence of sketched key poses. The user does not need any knowledge of the underlying mechanism used to create different expressions or facial poses, and no animation controls or parameters are directly manipulated. Instead, the user sketches the desired shape of a facial feature and the system reconstructs a 3D feature which fits the sketched stroke. This is achieved using a maximum likelihood framework where a statistical model in conjunction with Hidden Markov Models handles sketch detection, and a hierarchical statistical mapping approach reconstructs a posed 3D mesh from a low-dimensional representation.

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

1. Introduction

Animating a 3D face model with realistic expressions and facial poses generally requires extensive and skillful manual labour. The aim of facial animation research is to make it quick and easy to accurately pose an arbitrary 3D face model by offering high-level tools.

This paper shows how a sketching approach can be used to pose 3D face models using simple 2D strokes that project onto the 3D surface. We achieve this using a very small set of prior knowledge in the form of facial expressions and phonemes, which is used to fit a Gaussian mixture model that maps sketched strokes to facial features and poses.

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. Section 5 demonstrates how the system is used to create an animation sequence by sketching a series of keyframes. We then conclude with Section 6.

2. Background and previous work

Facial animation has been an active research topic since the work of Parke [Par72, Par74], where he parameterises facial expressions on a specific mesh and is able to create a range of expressions by varying the parameters. Since then, parameterised methods have proved successful. In 1978, Ekman and Friesen [EF78] developed the Facial Action Coding System (FACS) which has been incorporated by numerous researchers [Wat87, KMMtT91, CBK*06].

Blendshapes is the most popular facial animation technique used today. An artist creates key poses which are used to linearly interpolate new poses, where the blend can consist of whole faces or regional blends [PHL*98, LCF00]. Creating the key poses, sometimes called morph targets, needs skillful manual work unless they can be generated using motion data from real people [CB08]. Performance-driven methods where an actor performs the facial actions provide a more automatic and accurate way of generating realistic animations, where the actor's face is generally labelled with a set of markers. Deng and Neumann [ZD07] provide further information by describing a range of different facial animation techniques developed in recent years.

The combination of fusing together the performance-driven approach into an example-based technique is a recent trend in facial animation. Fundamentally, it gathers prior knowledge of facial movements by appling statistical inference on the motion data to achieve accurate reconstructions through maximum likelihood. Example-based sketch interface methods build on the same idea, but introduce an intuitive, high-level approach of controlling the facial poses through sketched strokes.

Chang and Jenkins [CJ06] looks for the optimal pose in a collection of key poses which they call articulation space. They do this using a reference and a target curve, and search for the optimal articulation weights which minimises the distance between the two curves, using a downhill simplex method. This collection can be either made up of blendshapes or alternatively articulation poses created using their their own approach. This is achieved by specifying particular regions of interest and applying various types of deformations on the mesh based on the curves and specified regions. This is not guaranteed to give realistic poses but is able to create new poses without any prior knowledge. Lau et al [LCXS07] further improve the notion of using a reference and a target curve to find the optimal pose in an space of pre-posed models by tackling the problem in a probability framework. The pre-existing models are treated as model priors used to find the posterior model which is the best match given the input strokes based on a mixture of factor analysers. In contrast, our method relies on predefined reference curves in the form of feature points, and asks the user only to sketch the target curves. This approach changes the way the interface is perceived. Instead of sketching changes, the notion is what you sketch is what you get. This simplifies the sketching process but it more limited outside the range of the pre-defined reference points which we call feature points (FPs). Sucontphunt et al [SMND08] use a different approach to posing a face model which aims to take the rigging process to a more intuitive level. Instead of manipulating the model itself in 3D, key points on a more simplified 2D sketch-based version are moved around to depict new poses which are then reconstructed in the 3D space. A prior knowledge gathered from motion data is used in a hierarchical Principal Components Analysis (PCA) model. This makes sure the reconstructed faces are realistic within the scope of the prior dataset. This method is efficient but currently the interactive sketch is limited to the front view. Deng and Neumann [ZD07] offer a more detailed description of the range of different facial animation techniques. Company et al [PCN05], and Olsen et al [OSSJ09] provide a more detailed survey on sketch-based interfaces.

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. The offline stages of our approach are as follows:

1. **Prepare facial poses -** The training data contains 36 poses of a 3D face mesh, each representing a different expression or viseme. Each pose is labelled with 46 feature points (FPs) giving two sets of corresponding poses, 'mesh poses' and 'FP poses' (Section 4.1).
2. **Construct a statistical model -** The FP training set is used to fit mixtures of probabilistic principal component analysers using the 3D FP coordinates (x,y,z) (see Section 4.2). The statistical model is used to analyse sketched strokes and to generate poses from incomplete data in the online stage.

The online stages of our approach are as follows:

1. **Interpret sketched strokes from a user -** The user sketches on the 3D face model where sketched points are mapped to the most likely FPs (Section 4.3).
2. **Find best pose -** A generative model uses the the FPs identified in 2 to find the remaining, unidentified FPs in order to make up a complete FP pose (Section 4.4).
3. **Reconstruct mesh pose from FP pose -** The complete FP pose acts as a set of control points used to deform the face mesh into the desired facial pose through a statistical mapping (Section 4.5).

4. Statistical model

4.1. Preparing training data of facial poses

The 36 face poses (neutral pose and 35 different expression and visemes poses) are created using FaceGen [†] where each mesh shares the same vertex topology. 46 feature points (FPs) are placed manually on the neutral pose using a subset of the MPEG-4 facial animation standard [Pak02]. The FPs are then mapped to the nearest vertex on the neutral mesh which is used to automatically calculate the FP coordinates for the remaining poses. Figure 1 shows the labelled FPs (in red) on the neutral pose and the anger pose without the eyes, tongue and teeth.

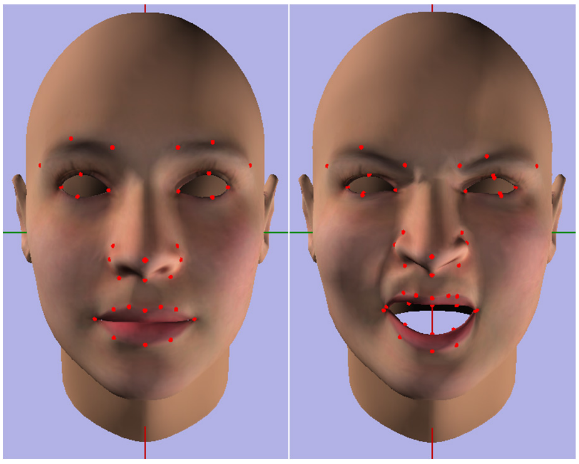

Figure 1: *Labelled FPs (shown in red) on the neutral pose (left), the anger pose (right). Note: Only the skin mesh is shown.*

[†] FaceGen; Singular Inversion

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 $\mathbf{x}$ with mean vector μ (where $d \gg q$) such that

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \mu + \varepsilon, \tag{1}$$

where ε is a Gaussian, independent noise model $\varepsilon \sim \mathbf{N}(0, \psi)$. This means that the observed vectors $\mathbf{t}$ are also Gaussian distributed, $\mathbf{t} \sim \mathbf{N}(\mu, \mathbf{C})$. 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 can be done either in closed form or using the EM algorithm as described by Michael and Bishop [TB99].

Using a single PPCA model to fit a univariate Gaussian function on the data set creates an unrealistic likelihood function. A multi-variate approach is needed to fit a non-linear data set. A Gaussian mixture model approximates a non-linear model by expressing the probability density function as a linear combination of basis functions

$$\mathbf{p}(\mathbf{t}) = \sum_{i=1}^{M} \pi_\mathbf{i}\mathbf{p}(\mathbf{t}|\mathbf{i}), \tag{2}$$

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

The parameters for this mixture model can be determined by maximising the data likelihood. For convenience, the problem is converted into an equivalent form where the goal is to minimise the negative log-likelihood which is treated like an error function. This cannot be calculated in closed form so the EM algorithm is employed to optimise the model parameters. Poorly initialised parameters can result in a local maxima problem as there are generally multiple local maxima of the log likelihood function. To reduce the chances of that happening the K-medoids and K-means methods are used to perform initial clustering [Bis07].

4.3. Finding FPs from sketched strokes

Every FP pose can be thought of as a low dimensional representation of its equivalent mesh pose. Figure 2 shows the FPs for every pose in the training set where points labelling the same feature form a cluster. The clusters are plotted with different colours to visualise the range of motion for each facial feature.

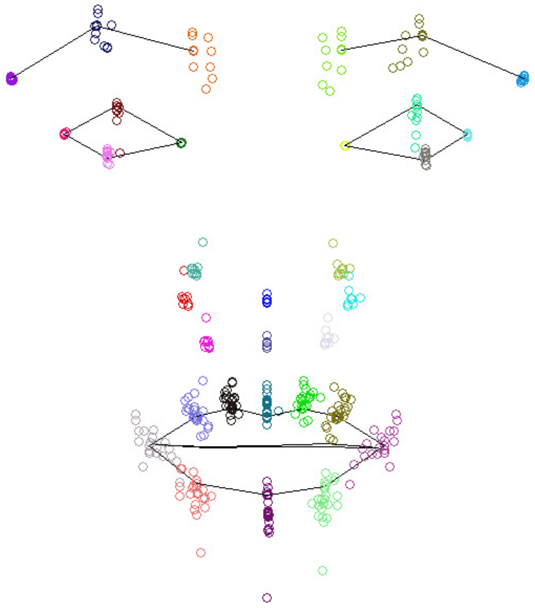

Figure 2: *Labelled FPs for every pose. Each FP cluster is shown as a different colour.*

The user sketches strokes representing the shape of facial features, e.g. whistling lips, sad eyebrows etc. A sketched stroke consists of a sequence of points where the assumption is they map to a corresponding sequence of FPs. The problem is finding this unknown sequence of optimal FPs describing the sketched feature based on an observed stroke. Instead of assigning sketched points to individual FPs independently, Hidden Markov Models (HMMs) are able to find the most probable sequence of hidden states (FPs) for a given observation sequence (stroke points).

The joint probability distribution over both the latent $\mathbf{Z}$ and observed variables $\mathbf{X}$ is

$$p(\mathbf{X}, \mathbf{Z}|\theta) = p(\mathbf{z}_1|\pi)\left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A})\right]\prod_{m=1}^{N} p(\mathbf{x}_m|\mathbf{z}_m, \phi), \tag{3}$$

where $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$, and $\theta = \{\pi, \mathbf{A}, \phi\}$ [Bis07]. θ contains the probability parameters where π describes the initial probabilities for each state (FP), $\mathbf{A}$ is the transition matrix which expresses the probability of moving from state to another, and ϕ is the emission probability which measures the probability of each stroke point belonging to each FP.

The emission probabilities are found by defining the FP clusters in a likelihood framework where the clusters naturally extend to the soft clustering approach embedded in the mixture model, and calculate the probability that a sketched point belongs to a particular FP cluster. A mixture model is fitted on 32 FPs (a subset of the labelled FPs), where each sample is the XYZ-coordinate of a single FP. Since there are 36 poses, the total number of samples in the training set is $32 * 36 = 1152$. (A subset is used to simplify the classification by omitting unnecessary FPs with regards to sketching. This includes the tongue, teeth and the FPs for the inner lip as they are not needed in most cases to distinguish between different lip poses.) Each group of FPs is defined as a cluster which means there are a total of 32 clusters (or mixture components), where the centre for each cluster is initialised as the mean of the corresponding FP coordinates.

When the user sketches a stroke, the marginal likelihood $\mathbf{p}(\mathbf{t_n})$ and the posterior responsibility

$$\mathbf{R_{ni}} = \frac{\mathbf{p}(\mathbf{t_n}|\mathbf{i})\pi_\mathbf{i}}{\mathbf{p}(\mathbf{t_n})} \tag{4}$$

is calculated for each point in the stroke. The responsibilities form a vector with 32 values, where each index represents a single cluster (see Figure 3). The values determine the probability of a single stroke point belonging to a particular cluster, where the values range from 0 to 1, and the sum of responsibilites is 1. This vector is used as the emission vector for the corresponding single stroke point in the emission matrix ϕ.

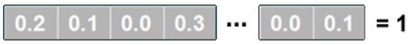

Figure 3: *Posterior responsibility for each cluster (total of 32 clusters) calculated for each sketched point.*

The initial probability matrix π is assigned with the probability values from ϕ which correspond to the first point on the stroke. However, strokes containing points which do not conform to the training set can cause problems. For instance when sketching an upper lip, the endpoints may not lie near a cluster likelihood range and will therefore be discarded. The endpoints carry important information as they define the boundaries of the desired feature. An example of this is if the stroke defining the upper lip is too short. If as a result the endpoints are ignored, the width of the lips will remain unchanged, while it is fairly likely that the user intended for them to become shorter. It is also possible that an endpoint has a lower likelihood than a neighbouring point on the stroke which maps to the same FP (lip corner). To help overcome this, it is assumed that users tends to draw a complete feature which contains the boundary points of the corresponding feature. For instance, a sketched upper lip will most likely contain the lip corners. Therefore, if a stroke's first point maps to a boundary FP, its initial probability is set to 1 and the initial probability for the remaining FPs are set to 0.

Entire strokes are sometimes ambiguous where they can be interpreted as more than one facial feature. An example of this is a sketched line that can either be an inner lower lip, or an outer lower lip. Also, a sketched eyebrow can be classified as the upper eyelid. This issue is dealt with in the initial probability matrix π by finding a suitable starting point, and the transition matrix $\mathbf{A}$ assures the stroke will not map to two distinct features by taking into account the semantic distance between FPs. This distance is determined by assigning FPs to different facial features such as upper lip, left eyebrow etc. Two FPs belonging to two different groups have a very high distance value which discourages unnatural jumping between features from one stroke point to another.

In order to find the most probable sequence of latent states (FPs) for a given set of observed data (strokes), the Viterbi algorithm [Vit67] is employed which is a max-sum algorithm whose complexity grows linearly with the length of the HMM trellis chain. The algorithm traverses through the HMM chain to find the optimal path through the trellis. The stroke point density is higher than the FP density which causes more than one point to be assigned to the same latent state. The FP (latent state) with the highest emission probability is assigned with the stroke point coordinates.

Other complications arise in a sketching interface that should be considered. The strokes are sketched in 2D which means the values along the depth axis based on the current viewpoint are unknown. The stroke points are projected onto the 3D model to get an estimate for the depth values in order to classify a set of FPs, but these projected values might not represent a realistic depiction of any known pose. Figure 4 demonstrates this problem where a particular lip shape is sketched from the front view (a) and then examined from a different viewpoint (b), where the FPs have been identified and assigned to stroke points. However, this particular 3D lip shape shown in (b) and (c) does not accurately describe any known pose. We want to be faithful to the sketched lip shape from the sketch-viewpoint which is the front view in this case. Therefore, we keep the coordinates on the axis plane seen from the front view (XZ), and update the values on the ambiguous depth axis seen from the front view (Y). The Y values for each identified FP are removed and replaced with the maximum likelihood values for the corresponding FP cluster, using the XZ values as conditional data. The updated coordinates are shown in (d) and contrasted with (c) where it primarily affected the lip corners. The reconstructed 3D lips are shown in (e) and (f) where the lips are now faithfully represented by the FPs.

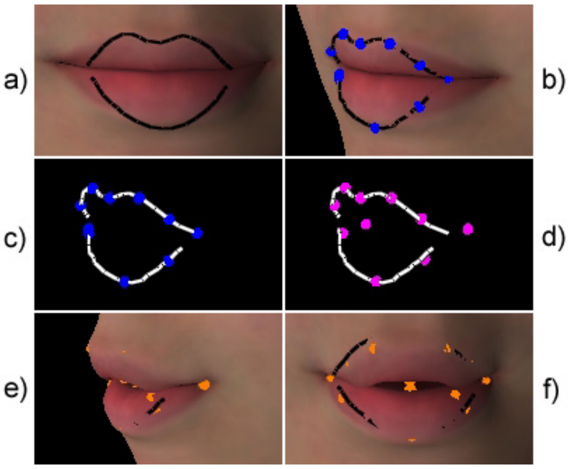

Figure 4: *Projected depth values for sketched strokes may not represent accurate values of the optimal pose.*

4.4. Reconstructing a complete pose

At this stage the system has identified sketched points mapping to a number of FPs from the set of 46 FPs describing a single facial pose. There are typically a large number of unidentified FPs as the user is not expected to draw every aspect of the pose. In addition to that there are 14 FPs that cannot be sketched here (teeth, tongue and inner lips) which means the FP pose is always incomplete. A generative probability model is needed to find the most likely pose given a partial pose defined by the FPs that were identified from the sketched strokes.

K mixtures are fitted on a training set consisting of $n = 36$ different poses, where each pose contains 46 3-dimensional FPs. A single training sample is therefore a vector with $46 * 3$ dimensions, making up a training set of size 36×138. The FPs identified from a set of sketched strokes are used as observed data where the max-conditional distribution is calculated over the missing points to construct a complete FP pose. This is done by partitioning the data into observed data *(o)* and missing data *(m)* (see Figure 5). where the observed data consists of the classified FPs acquired in Section 4.3.

The $\mathbf{t}$ is the sample vector for the pose made of up the observed and missing data which trivially form the observed and missing partitions. The same partitioning is applied to the mean pose μ and covariance $\mathbf{C}$ stored in the mixture model. The expected values for the missing data $\mathbf{t_m}$ are found using the conditional distribution $\mathbf{p}(\mathbf{t_o}|\mathbf{t_m})$ where

$$\begin{aligned}
&\mathbf{t} = (\mathbf{t_m};\mathbf{t_o})\\
&\mu = (\mu_{\mathbf{m}};\mu_{\mathbf{o}})\\
&\Lambda = (\Lambda_{mm}\Lambda_{mo};\Lambda_{om}\Lambda_{oo})\\
&\Lambda_{mm} = (\mathbf{C_{mm}} - \mathbf{C_{mo}}\mathbf{C_{oo}}^{-1}\mathbf{C_{om}})^{-1}\\
&\Lambda_{mo} = -(\mathbf{C_{mm}} - \mathbf{C_{mo}}\mathbf{C_{oo}}^{-1}\mathbf{C_{om}})^{-1}\mathbf{C_{mo}}\mathbf{C_{oo}}^{-1}\\
&\mathbf{t_m} = \mu_{\mathbf{m|o}} = \mu_{\mathbf{m}} - \Lambda_{mm}^{-1}\Lambda_{mo}(\mathbf{t_o} - \mu_{\mathbf{o}}),
\end{aligned} \tag{5}$$

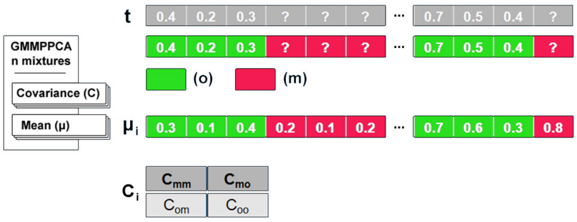

Figure 5: *Partitioning the FPs into observed (conditional) and missing data: o=observed; m=missing;*

and $\Lambda \equiv \mathbf{C}^{-1}$ is known as the precision matrix. *oo*, *om*, *mo*, and *mm* correspond to the combinations of partitioning the observed and missing [row][column] entries in the matrices Λ and $\mathbf{C}$. This is done for every mixture component $k = 1..K$ using the corresponding mean and covariance. The complete pose for k is found by concatenating the observed data with the expected data, and is referred to as a reconstructed FP pose. The probability ρ_k is then calculated using

$$\begin{aligned}
&\mathbf{p}(\mathbf{t}) = (2\pi)^{-d/2}|\mathbf{C_i}|^{-1/2}\exp\{-\tfrac{1}{2}(\mathbf{t}-\mu_{\mathbf{i}})^T\mathbf{C_i^{-1}}(\mathbf{t}-\mu_{\mathbf{i}}))\}\\
&\rho_k = \mathbf{p}(\mathbf{t}) * \pi_{\mathbf{i}}{}^T.
\end{aligned} \tag{6}$$

The reconstruction tied with $max(\{\rho_1,..,\rho_K\})$ is selected and is used to create the mesh pose using the method in the next section.

The following examples show how well the selected generative model estimates a complete pose based on incomplete input data for a particular mixture model. Figure 6 shows the reconstructed pose when using three observed FPs (green) describing the left eyebrow taken from the anger pose (red x's). The blue circles represent the reconstructed pose which lies very close to the target pose despite the small amount of observed data. Figure 7 shows a similar scenario for an open smile, but now only one observed FP is used describing the right lip corner. The model is able to predict accurately that the intention is to create an open smile.

4.5. Reconstructing mesh pose from FP pose

The FPs act as control points which are used to deform the mesh vertices in order to create a range of different facial poses. The set of FP poses are referred to as P, where $p_i \; \varepsilon \; P, i = 1..n$ is a single FP pose, and similarly the set of mesh poses as V, where $v_i \; \varepsilon \; V$ is a single mesh pose. A statistical mapping $\Psi : P \rightarrow V$ is defined which maps a d_P-dimensional FP vector to a d_V-dimensional vertex vector. Figure 8 visualises the mapping process where the left side contains the FPs, and the right side contains the mesh vertices where $n = 36$.

Because $d_V \gg n$, a dual approach to PPCA is performed on both P (left) and V (right) where instead of marginalising the latent variables $\mathbf{X}$ and optimising the parameters $\mathbf{W}$ via

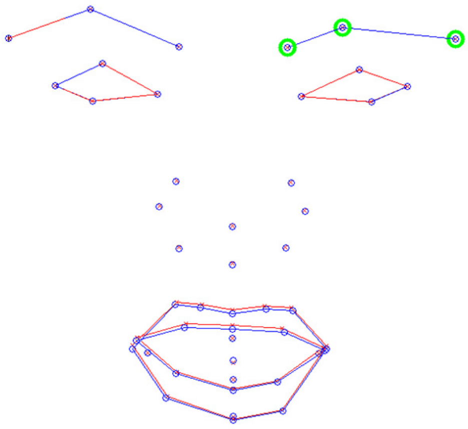

Figure 6: *Reconstructing anger pose from incomplete data. FPs circled in green are the observed data. Red x's represent the target pose. Blue circles show the expected data.*

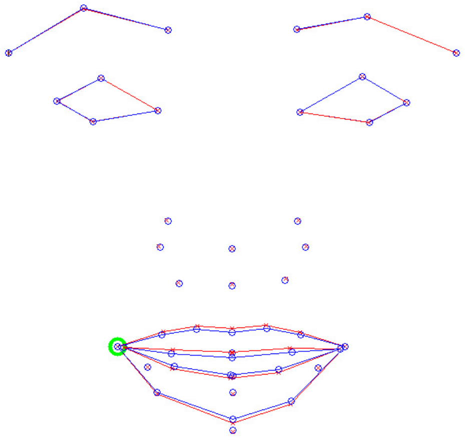

Figure 7: *Reconstructing open smile pose from incomplete data. FPs circled in green are the observed data. Red x's represent the target pose. Blue circles show the expected data.*

maximum likelihood, the parameters are marginalised and optimising with respect to the latent variables [Law05]. The latent variables $\mathbf{x}$ for each pose training sample $\mathbf{t}$ are calculated for both P and V to form $n \times q_P$ and $n \times q_V$ latent matrices using

$$\begin{aligned} \mathbf{M}^{-1} &= \sigma^2\mathbf{I} + \mathbf{W}^T\mathbf{W}, \\ \mathbf{W} &= \mathbf{U_S}(\Lambda - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{R}, \\ \sigma^2 &= \tfrac{1}{d-q}\textstyle\sum_{j=q+1}^{d}\lambda_j, \\ \mathbf{x} &= \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{t} - \mu), \end{aligned} \tag{7}$$

where q is either q_P or q_V, Λ is a $q \times q$ diagonal matrix containing the q eigenvalues $\lambda_1, .., \lambda_q$, and $\mathbf{R}$ is an arbitrary $q \times q$ orthogonal rotation matrix [TB99].

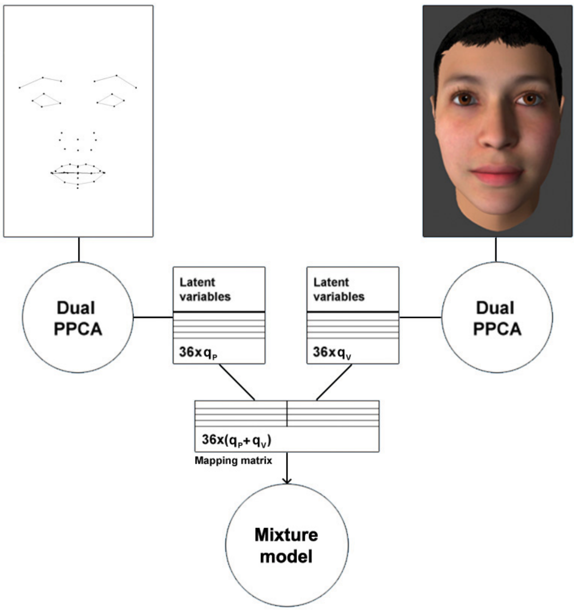

Figure 8: *Mapping FP pose to a mesh pose.*

We then join the two matrices together to form a $36 \times (q_P + q_V)$ matrix which is our mapping matrix and is learned using a Gaussian mixture model. Given a set of FPs, the vertex structure for a corresponding face mesh can be found by calculating the latent variables for the FPs using Equation 7, finding the missing vertex latent variables using Equation 5, and reconstruct the full mesh η where

$$\eta = \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{M}\mathbf{x} + \mu. \tag{8}$$

To verify this method is capable of producing the correct mesh pose structure from only 46 feature points, the 36 target mesh poses (V) are mapped from the corresponding set of feature points (P). Figures 9 and 10 show the reconstruction of two facial poses used throughout this paper, the anger pose and the open smile respectively. The upper left corner shows the FPs extracted from the target pose, and the bottom left and right show the mesh reconstruction. The red dots represent the original target vertices for the given pose, and the blue circles display the reconstructed vertices. A reconstruction for a particular vertex has zero error if the red dot lies perfectly within the centre of its corresponding circle.

The next section shows how using the statistical mapping, a whole range of facial expressions can be generated using only 36 target poses as a training set. The system is capable of reconstructing every target mesh pose, as well as a gradient of poses not present in the training set.

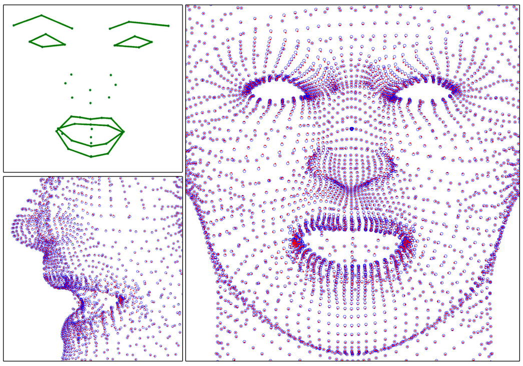

Figure 9: *Anger pose. Reconstruction using the marked feature points for the pose. Blue circles show the reconstructed mesh, and the red dots specify the target mesh for this pose.*

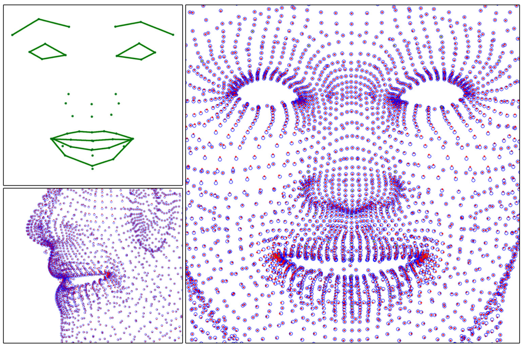

Figure 10: *Open smile pose. Reconstruction using the marked feature points for the pose. Blue circles show the reconstructed mesh, and the red dots specify the target mesh for this pose.*

5. Creating an animation using sketched keyframe poses

As a starting point, the user is presented with the neutral pose where he can sketch directly on the model from any viewpoint. When the user is satisfied the system identifies observed FPs from the sketched points using the method described in Section 4.3. The expected values for the remaining FPs are calculated (Section 4.4), and used to recover an updated vertex structure (Section 4.5). The user can continue sketching to make further adjustments until he is satisfied with the pose and adds it into a sequence of keyframes. Figure 11 shows some keyframes produced with some simple strokes starting with the neutral pose in the top left corner. Intermediate frames are rendered to make up a complete animated sequence. This could be done by linearly interpolating between two keyframe models, but instead the keyframe FPs are interpolated using a cardinal spline and the generative model is used to reconstruct the model for each frame. This is done to prove the system can generate a gradient of facial poses which are accurate enough to generate a smooth motion for every facial feature. Figure 12 shows two keyframes and 10 intermediate frames producing an animated sequence using the interpolated FPs as input for each frame.

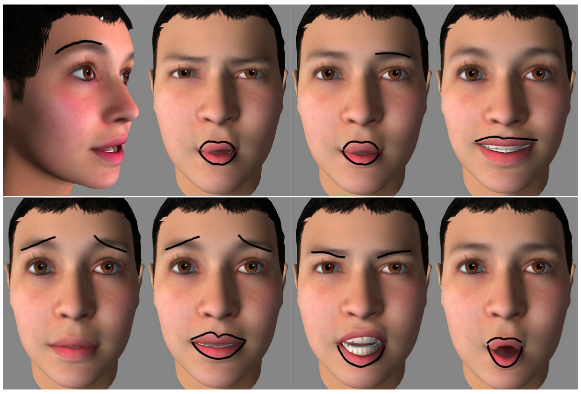

Figure 11: *Sketching keyframes.*

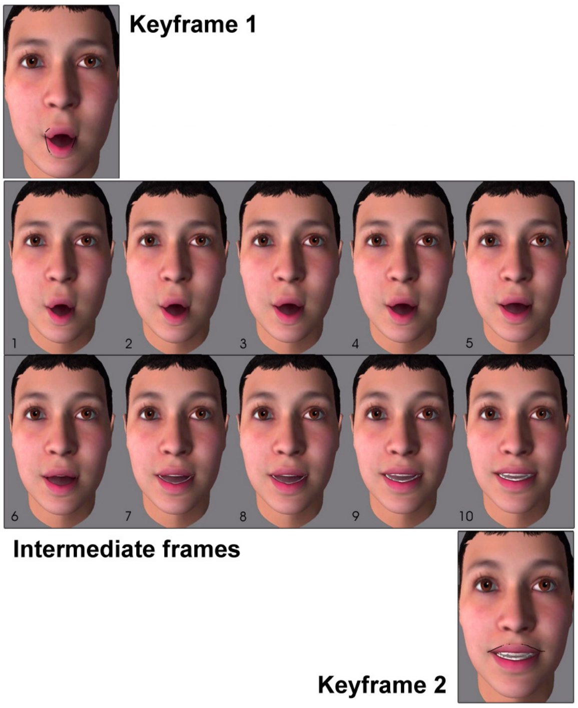

Figure 12: *Two keyframes and 10 generated intermediate frames. An animation file for this example is available.*

6. Conclusions and future work

We have presented a new approach to creating 3D facial animation through sketching. Sketching acts as a high-level control to modelling where a new pose can be created by indicating the desired outcome as opposed to applying animation targets, moving individual control points, or tweaking semantic parameters. Very few sketch strokes are needed to construct a new pose through incomplete data handling. We accomplish this using a knowledge-base in the form of a statistical model that through a maximum likelihood approach knows how poses are constructed from partial input. The input is made up of FPs describing each pose in a low-dimensional space. Modifying one aspect of the face automatically correlates other areas on the face to match accordingly. Facial expressions generated using this system are therefore always complete and plausible.

Using only 36 poses as training data we are able to create a large range of facial poses, and accurately calculate intermediate frames. However with a more extensive data set the results could be further improved along with the addition of new poses that fall outside the likelihood range of our current system. The system is limited to making changes to areas that have pre-defined FPs. However the FPs classify all the main facial features and the correlation makes sure every area on the face is adapted to match the desired pose.

References

[Bis07] BISHOP C. M.: *Pattern Recognition and Machine Learning*. Springer, 2007. 3

[CB08] CHUANG E., BREGLER C.: Performance driven facial animation using blendshape interpolation. 1

[CBK*06] CURIO C., BREIDT M., KLEINER M., VUONG Q. C., GIESE M. A., BH. H.: Semantic 3d motion retargeting for facial animation. In *APGV '06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization* (New York, NY, USA, 2006), ACM, pp. 77–84. 1

[CJ06] CHANG E., JENKINS O. C.: Sketching articulation and pose for facial animation. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 271–280. 1

[EF78] EKMAN P., FRIESEN W.: *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, 1978. 1

[KMMtT91] KALRA P., MANGILI A., MAGNENAT-THALMANN N., THALMANN D.: Smile: A multilayered facial animation system. In *In T.L. Kunii, editor, Modeling in Computer Graphics* (1991), Springer-Verlag, pp. 189–198. 1

[Law05] LAWRENCE N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research 6* (2005), 1783–1816. 6

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 165–172. 1

[LCXS07] LAU M., CHAI J., XU Y.-Q., SHUM H.-Y.: Face poser: interactive modeling of 3d facial expressions using model priors. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 161–170. 2

[OSSJ09] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A.: Technical section: Sketch-based modeling: A survey. *Comput. Graph. 33*, 1 (2009), 85–103. 2

[Pak02] PAKSTAS A.: *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2002. 2

[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. 1

[Par74] PARKE F. I.: *A parametric model for human faces.* PhD thesis, 1974. 1

[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. 2

[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. 1

[SMND08] SUCONTPHUNT T., MO Z., NEUMANN U., DENG Z.: Interactive 3d facial expression posing through 2d portrait manipulation. In *GI'08: Proc. of Graphics Interface* (Windsor, Ontario, Canada, 2008). 2

[TB99] TIPPING M. E., BISHOP C. M.: Mixtures of probabilistic principal component analysers. *Neural Computation 11*, 2 (1999), 443–482. 3, 6

[Vit67] VITERBI A. J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory 13* (1967), 260–269. 4

[Wat87] WATERS K.: A muscle model for animation three-dimensional facial expression. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM, pp. 17–24. 1

[ZD07] ZHIGANG DENG U. N.: *Data-Driven 3D Facial Animation*, 1st ed. Springer, 2007. 1, 2