Face Modeling and Editing with Statistical Local Feature Control Models

Yu Zhang[*] and Norman I. Badler

Department of Computer and Information Science

University of Pennsylvania

Abstract

This paper presents a novel method based on statistical facial feature control models for generating realistic controllable face models. The local feature control models are constructed based on the exemplar 3D face scans. We use a three-step model fitting approach for the 3D registration problem. Once we have a common surface representation for examples, we form feature shape spaces by applying a principal component analysis (PCA) to the data sets of facial feature shapes. We compute a set of anthropometric measurements to parameterize the exemplar shapes of each facial feature in a measurement space. Using PCA coefficients as a compact shape representation, we approach the shape synthesis problem by forming scattered data interpolation functions that are devoted to the generation of desired shape by taking the anthropometric parameters as input. The correspondence among all exemplar face textures is obtained by parameterizing a 3D generic mesh over a 2D image domain. The new feature texture with desired attributes is synthesized by interpolating the exemplar textures. With the exception of an initial tuning of feature point positions and assignment of texture attribute values, our method is fully automated. In the resulting system, users are assisted in automatically generating or editing a face model by controlling the high-level parameters.

Key Words: Face Modeling, Facial Features, Anthropometry, Statistical Estimation, Model Fitting, Interpolation, 3D Scanned Data.

1 Introduction

The human face is a key component of human interaction and communication. For this reason generation of realistic face models has been one of the most interesting problems in computer graphics. At present, a variety of face modeling methodologies are available, which can be classified into the creative approach and the reconstructive approach. The creative approach is to facilitate manual specification of the new face model by a user. A certain facility has been offered already by commercial modelers; some researchers have also sought to provide parameterized models for interactive user control [26, 54, 59, 66, 68]. This approach provides full

[*]Corresponding author. Tel.: 215-573-3013; fax: 215-573-7453; Email: yuzha@seas.upenn.edu

control over the result, including the ability to produce cartoon effects and the high efficiency of geometric manipulation. However, it requires a great deal of expertise to avoid unrealistic results.

The reconstructive approach is to extract face geometry from the measurement of a living subject. The most accurate method is to make a plaster model of a face and to scan this model using a precise laser range system. However, not everybody can afford the considerable time and expense this process requires. In addition, the molding compound may lead to sagging of facial features. With a significant increase in the quality and availability of 3D capture methods, a common approach towards creating face models of real humans uses laser range scanners (e.g., CyberWare) to acquire both the face geometry and texture simultaneously [41, 42, 53]. Although the acquired face data is highly accurate, the scanned model corresponds to a single individual that tells us little about the spaces of face shapes and textures; and each new face must be found on a subject. The user does not has any control over the model to edit it in a way that produces a plausible, novel face.

In this paper a new method is presented to **bridge the creative approach and reconstructive approach**. Our method estimates explicit high-level control models of human faces by examining geometry and color variations among captured face data. Since internal facial features (e.g. eyes, nose, mouth and chin) are good for discriminating faces [87], we construct a face synthesis system by perceiving the face as a set of feature regions. The feature-based synthesis allows us to generate more diverse faces through various combinations of synthesized features. The learned control models define a mapping from control parameters back to facial feature shape or texture, thus we can use them to automatically synthesize varied geometric models of human faces. As a result, our system provides an intuitive interface for users via controllable attributes of facial features. And because the control models are estimated based on the extracted statistics from the example models, the output model of the system maintains the quality that exists in the real faces of individuals.

Our method takes as examples 3D face scans from a large database. We use a three-step model fitting approach for the 3D registration problem, where the fitting of a generic mesh to each example is carried out in a global-to-local fashion. By bringing scanned models into full correspondence with each other, we are able to apply a principal component analysis (PCA) to the exemplar shapes of each facial feature to build a low-dimensional shape space. We characterize and explore the space of probable feature shapes using high-level control parameters. We parameterize the example models using anthropometric measurements, and predefine the interpolation functions for parameterized example models based on radial basis functions. At runtime, the interpolation functions are evaluated to efficiently generate the appropriate feature shapes by taking the anthropometric parameters as input. We automatically determine correspondence among exemplar face textures by constructing a parameterization of the 3D generic mesh over a 2D image domain. Having the texture interpolators formulated, the runtime texture synthesis is reduced to the interpolation function evaluation according to the texture attribute parameters.

The resulting system can be used for many scenarios. For applications in art/design, computer artists can use our system to create face models as desired for virtual characters in computer games, films and advertisement, human-computer interfaces, and for avatars in the virtual reality space. This system also has the potential application in forensic sciences, where composite face models/images of suspects need to be generated from descriptions given by the eyewitnesses. Both cases involve the design of new natural looking faces that can occur in the real world or the manipulation of an existing face to emphasize the characteristics of the face.

The main contributions presented in this paper are:

- A general, controllable, and practical system for face modeling. It estimates high-level control models in order to infer a particular face from intuitive input controls. In principle, given a large and varied data set, the control models can generate any face.
- As correlations between control parameters and the face shape and texture are estimated by exploiting the real faces of individuals, our method regulates the naturalness of synthesized faces. Unspecified parts of the synthesized facial features are automatically completed according to statistical properties.
- We propose a new algorithm which uses intuitive attribute parameters for navigating face space. Particularly, our system provides sets of comprehensive anthropometric parameters to easily control face shape characteristics by taking into account the physical structure of real faces.
- A robust model fitting approach for establishing correspondences between scanned surfaces.
- The automatic runtime face synthesis is efficient in time complexity and performs fast.

The outline of this paper is as follows. Section 2 reviews the previous and related work. Section 3 provides a functional overview of our 3D face modeling system. Section 4 describes preparation of the generic model and example models from the 3D scanner. Section 5 details the model fitting process. Section 6 describes our technique of feature-based shape synthesis. The approaches to mesh parameterization, local texture synthesis and texture blending are described in Section 7. Experimental results are shown in Section 8. Section 9 presents conclusions and proposes avenues for future work.

2 Previous and Related Work

Face modeling and animation is an active area of research in computer graphics. There have been extensive efforts on the development of 3D facial animation techniques (see [63, 65] for a good overview). Beginning with Parke's pioneering work [67], desire for improved realism has driven researchers to extend geometric models [26, 59, 61, 66] with physically-based models which attempt to model the influence of muscle contraction onto the skin surface by approximating the biomechanical properties of skin [41, 42, 47, 53, 69, 79, 83, 92]. Physically-based models were combined with non-linear finite element methods in systems that could be used for planning facial surgeries [44, 46]. Free-form deformations have been employed by Kalra et al. [43] to manipulate facial expressions. In parallel, Williams presented a compelling argument [84] in favor of performance-driven facial animation, which anticipated techniques for tracking head motions and facial expressions in video [29, 56, 70]. In performance-driven approaches, colored markers painted on the face or lips are extensively used to aid in tracking facial motion [33, 35]. A more expensive alternative could use a 3D scanning technique [88], if the performance can be re-recorded with such a system.

Other recent work includes facial animation based on audio or text input [9, 14, 18, 28], learned morphable models for animating faces in images and video [7, 6] and for exchanging faces in images [8], geometry-driven facial expression synthesis [89], facial motion transfer [24, 60, 62, 73, 81], wrinkle formation [86, 91], eye motion [25, 51], automatic estimation of facial muscle contractions [17, 76, 77], and biomechanical neck modeling [50]. Data-driven speech animation approaches concatenate phoneme or syllable segments [12, 48, 58] or modeling speech co-articulation from data [22, 23, 45]. Byun and Badler [11] modify the MPEG-4 Facial Animation Parameters (FAPs) to add expressiveness. Chai et al. [15] use facial tracking to drive animations from a motion capture database. Wang et al. [82] use a multiresolution deformable mesh to track facial motion, and a low dimensional embedding technique to learn expression style.

Regarding modeling of static face geometry in particular, several approaches are documented in the literature. Parametric conformation models have been invented very early [26, 54, 59, 66, 68]. The desire was to create an encapsulated model that could generate a wide range of faces based on a small set of input parameters. However, the parameterizations have been developed without careful attention to the physical structure of the face. The choice of the parameter set depends on the face mesh topology and therefore the manual association of a group of vertices to a specific parameter is required. Furthermore, manual parameter tuning without constraints from real human faces for generating a realistic face is difficult and time-consuming.

The image-based technique [1, 34, 35, 39, 49, 52, 57, 64, 71] utilizes an existing 3D face model and information from few pictures (or video streams) for the reconstruction of both geometry and texture. Although this kind of technique can provide reconstructed face models easily, its drawbacks are the inaccurate geometry reconstruction and inability to generate new faces that have no image counterpart. Another limiting factor of this technique lies in that it hardly gives any control to the user.

The modeling of facial variations has been attempted based on physical measurement statistics and the physical aspects of facial deformation. Decarlo et al. [20] construct a range of face models with realistic proportions using a variational constrained optimization technique. However, because of the sparseness of the constraints compared to the high dimensionality of possible faces, realistic shape cannot be obtained in the facial regions where no desirable measurement has been explicitly specified as a constraint. Also, this approach requires minutes of computation for the optimization process to generate a face. In addition, variations in face texture are ignored in their work. In contrast, our method is efficient to generate a new face with desired shape and texture within a second. Moreover, we utilize the prior knowledge of the face shape in relation with the given measurements to regulate the naturalness of modeled faces, maintaining the quality that exists in the real faces of individuals. Kähler et al. [41] use statistic data of face anthropometric measurements to drive the landmark-based face deformation according to growth and aging.

Blanz and Vetter [6] present a process for estimating the shape of a face in a single photograph, and a set of controls for manipulation of appearance attributes. There are several key differences from our work. First, they manually assign the attribute values to face shape and texture and devise attribute controls for single variable using linear regression. We automatically compute the anthropometric measurements for face shape and relate several variables simultaneously by learning a mapping between the measurement space and the shape space through scattered data interpolation. Second, they use a 3D variant of a gradient-based optical flow algorithm to derive the point-to-point correspondence. This approach will not work well for faces of different races or in different illumination given the inherent problem of using static textures. We present a robust method of determining correspondences that does not depend on the texture information. Third, our method guarantees local control thanks to its feature-based nature. An interface for gradient-based face space navigation has been proposed in [16]. Principal components that are not intuitive to users are used as navigation axes in face space, and facial features cannot be controlled individually. The authors focus on a comparison of different user interfaces.

The composite facial images are the basis of some approaches to retrieving an image from a mug-shot database. The database used in [85] is indexed by attributes of individual features obtained by PCA. The best matches to a photofit target image are retrieved based on normalized correlation and weighted distance of the feature vectors. Both Brunelli and Mich [10] and Baker and Seltzer [5] propose systems that use facial composites for database retrieval and PCA for computation of similarity. Several commercial systems for generating composite facial images are available [72, 27, 38]. Although they are effective to use, 2D face composite still lacks some of the advantages of a 3D model, such as the complete freedom of viewpoint and the ability to be combined with other 3D graphics. Additionally, to our knowledge, no commercial 2D composite system available today supports automatic completion of unspecified facial regions according to statistical properties.

Example-based synthesis is another stream of research related to our method. Rose et al. [74] and Sloan et al. [78] propose example-based motion blending frameworks, employing scattered data interpolation with radial basis functions. Lewis et al. [55] introduce an example-based pose space deformation technique, and Allen et al. [3] apply a similar technique to range scan data for creating a new pose model. More recently, Allen et al. [2] and Seo et al. [75] present methods for generating a variety of human body shapes with intuitive parameters based on unorganized scanned data. However, the gross body and face are different significantly in structure and representation. The gross body shape usually varies globally and can be compactly parameterized by a relatively small set of key measurements such as the girth and length of the body parts. Body color attributes are typically those that provide a global description and are usually related to the variable of race. On the other hand, the human face is an extremely complex geometric form, consisting of a diverse and irregular structure. The delicate face structure requires far more measurements for its description. Moreover, the face exhibits various textures due to presence of the moustache, eyebrows and wrinkles as well as variations in skin color. Therefore, modeling the face requires different level of accuracy and realism compared to other parts of human body.

FaceGen 3 [30] is an existing system that we have found to be similar to ours in functionality. However, there is not much information available about how this function is achieved since it is a commercial product. As far as we know, it does not split the face mesh into different independent regions for truly localized deformation. As a consequence, this system is more liable to produce global face variations. In contrast, we use a region-based method for local face shape control. The controls of different face regions are *orthogonal* to each other, and form a meaningful basis for the face space - the control basis can produce the set of more diverse face shapes.

3 System Overview

A block diagram of the system architecture is illustrated in Fig. 1. It consists of an offline processing unit and a runtime unit.

The offline processing makes use of a generic model and 3D scanned models. Using the same generic model for fitting to each scanned model makes trivial the feature correspondence among the scanned models. Feature correspondence is then used to form a low dimensional local shape space of each facial feature. We use the face anthropometric measurements to parameterize the facial feature shape of the examples in a measurement space. The shape synthesizer generates new facial feature geometries using anthropometric measurements as input parameters. It is derived by determining the mapping from the control parameters to the facial feature geometry through an RBF-based interpolation network. For synthesizing local facial textures, the mesh parameterization procedure prepares the example textures in mutual correspondence. Similar to the shape synthesizer, texture constraints derived from the statistics of scanned models are used to guide local texture synthesis. At runtime,

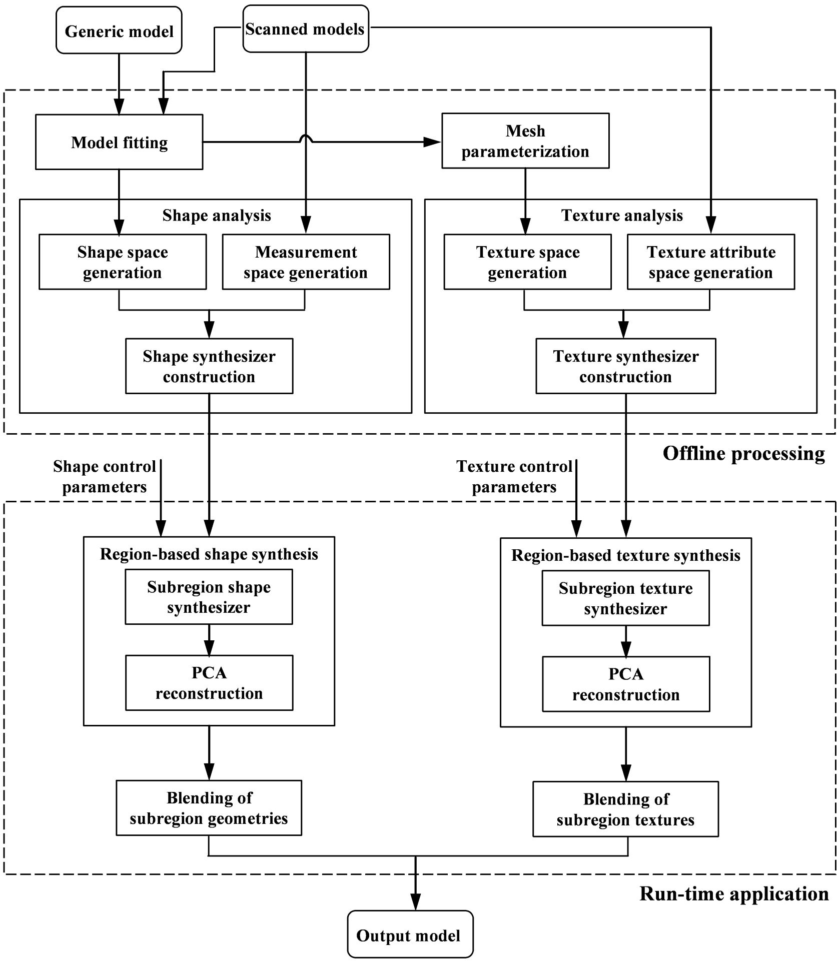

Generic model
Scanned models
Model fitting
Mesh parameterization
Shape analysis
Shape space generation
Measurement space generation
Shape synthesizer construction
Texture analysis
Texture space generation
Texture attribute space generation
Texture synthesizer construction
Offline processing
Shape control parameters
Texture control parameters
Region-based shape synthesis
Subregion shape synthesizer
PCA reconstruction
Region-based texture synthesis
Subregion texture synthesizer
PCA reconstruction
Blending of subregion geometries
Blending of subregion textures
Run-time application
Output model

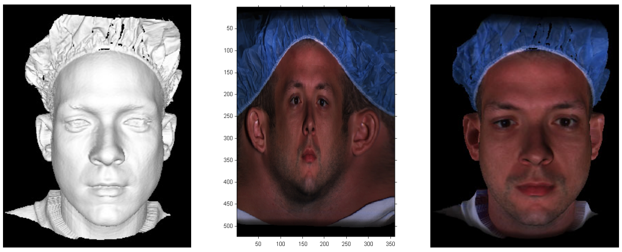
Figure 1: Overview of the data-driven 3D face synthesis system.

the system generates new geometries and textures of individual features from the input parameters by evaluating the constructed synthesizers and PCA models.

4 Face Data

We use the USF face database [80] that contains Cyberware face scans of 186 subjects with various age, gender and race. The age of the subjects ranges from 17 to 68 years, and there are 126 male and 60 female subjects. Most of the subjects are Caucasians (129), with African-Americans making up the second largest group (37), and Asians the smallest group (20). All faces are without makeup and accessories. Each subject is captured wearing a bathing cap and with a neutral expression. The laser scans provide face structure data which contains approximately 80,000 surface points (see Fig. 2 (a)) and RGB-color values that are stored in a 360× 524 image with 8 bit per channel for texture-mapping (see Fig. 2 (b)(c)).

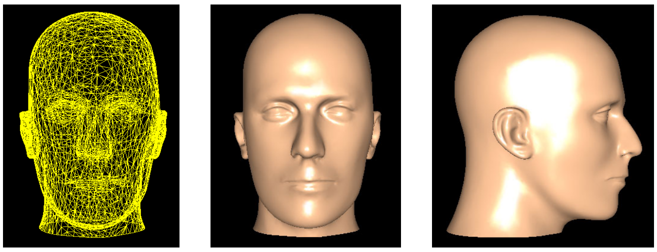
We use a generic head model created with Autodesk Maya [4]. It is a wire-frame of numbered vertices in 3D coordinate space (see Fig. 3 (a)). Since the triangle (the 2D simplex) offers the greatest flexibility in tessellating 2D manifolds in the 3D Euclidean space, our head model uses a triangular mesh, eliminating the topological restrictions of rectangular meshes. It consists of 1,092 vertices and 2,274 triangles. Prescribed colors are added



(a) (b) (c)

Figure 2: Face data: (a) scanned face geometry; (b) acquired color image; (c) texture-mapped face scan.

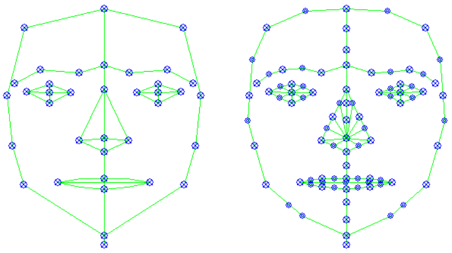
to each triangle to form a smooth-shaded surface (see Fig. 3 (b) and (c)).



(a) (b) (c)

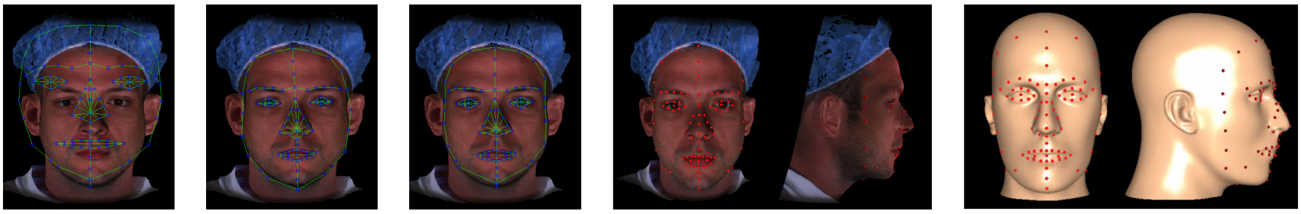
Figure 3: Generic model: (a) wire-frame mesh; (b) and (c) smoothly shaded surface.

Let each 3D face in the database be $\mathcal{F}_i$ ($i = 1, \ldots, M$). Since the number of vertices in $\mathcal{F}_i$ varies, we resample all faces in the database so that they have the same number of vertices all in mutual correspondence. Feature points are identified to guide the resampling. The number of feature points on the face model, their distribution, and their association with points on the face model are important to achieve high quality model fitting. A feature point is associated with a single point of the scanned model. For the generic model represented by much fewer vertices, the landmark is instead associated with the barycentric coordinates of one triangle.



(a) (b)

Figure 4: 2D feature mask: (a) initial set of feature points; (b) after additional feature points have been inserted.



(a) (b) (c) (d) (e)

Figure 5: Semi-automatic feature point identification: (a) initial outline of the feature mask; (b) after automatic facial feature detection; (c) after interactive tuning; (d) and (e) feature points identified on the scanned data and generic model.

As illustrated in Fig. 4 (a), a 2D *feature mask* consisting of polylines groups a set of 37 feature points that correspond to the facial features. However, because of the limited number of initial feature points, some undersampled regions could require additional information in order to improve the shape approximation for the fitted generic model. This is achieved by introducing additional correspondences. Each polyline in the feature mask is subdivided with new feature points inserted by stepping along it using an arc-length parametrization, keeping in mind that each new feature point increases the computation of LU decomposition and deformation of the generic model through the RBF-based interpolation as will be described in Section 5.1. We then run the scattered data interpolation algorithm to update the vertices of the generic model for evaluation. After refinement, the final feature mask consists of 86 feature points (see Fig. 4 (b)). A further increase in the number of the feature points would not produce an appreciable improvement in the quality of model fitting.

In our method, the 3D feature points are identified semi-automatically. Fig. 5 depicts the process. The refined feature mask is superimposed onto the front-view face image obtained by orthographic projection of a textured 3D face scan into an image plane. The facial features in this image are identified by using the Active Shape Models (ASM) [19] and the feature mask is fitted to the features automatically. The feature mask can be manipulated interactively. A little user interaction is needed to tune the feature point positions due to slight inaccuracy of the automatic facial feature detection. The 3D positions of the feature points on the scanned surface are then recovered by back-projection to 3D space. In this way, we efficiently define a set of feature points in a face $\mathcal{F}_i$ as $U_i = \{\mathbf{u}_{i,1}, \ldots, \mathbf{u}_{i,n}\}$, where $n = 86$. Our generic model $\mathcal{G}$ is already tagged with the corresponding set of feature points $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ by default.

5 Model Fitting

5.1 Global Warping

The problem of deriving full correspondence for all models $\mathcal{F}_i$ can be stated as: resample the surface for all $\mathcal{F}_i$ using $\mathcal{G}$ under the constraint that $\mathbf{v}_j$ is mapped to $\mathbf{u}_{i,j}$. The goal is to construct a smooth interpolating function that expresses the deformation of the non-feature skin vertices in terms of the changes in the feature points during warping. This problem is addressed by scattered data interpolation based on Radial Basis Functions (RBFs). The displacement vector $\mathbf{d}_{i,j} = \mathbf{u}_{i,j} - \mathbf{v}_j$ is known for each feature point $\mathbf{v}_j$ on the generic model and $\mathbf{u}_{i,j}$ on the scanned surface. These displacements are utilized to construct the interpolating function that returns the displacement for each generic mesh vertex:

$$\mathbf{f}(\mathbf{x}) = \sum_{j=1}^{n} \mathbf{w}_j \phi_j(\|\mathbf{x} - \mathbf{v}_j\|) + \mathbf{M}\mathbf{x} + \mathbf{t} \tag{1}$$

where $\mathbf{x} \in \mathcal{R}^3$ is a vertex on the generic model, $\|\cdot\|$ denotes the Euclidean norm and ϕ is a radial basis function. $\mathbf{w}_j$, $\mathbf{M}$ and $\mathbf{t}$ are the unknown parameters. Among them, $\mathbf{w}_j \in \mathcal{R}^3$ are the interpolation weights, $\mathbf{M} \in \mathcal{R}^{3\times3}$ represents rotation and scaling transformations, and $\mathbf{t} \in \mathcal{R}^3$ represents translation transformation.

Different functions for $\phi(r)$ are available. Some pros and cons are discussed by Carr et al. [13]. We had better results visually with the multi-quadric function $\phi(r) = \sqrt{r^2 + \rho^2}$, where ρ is the locality parameter used to control how the basis function is influenced by neighboring feature points (see Fig. 6). ρ is determined as the Euclidean distance to the nearest other feature point. To determine the weights $\mathbf{w}_j$ and the affine transformation parameters **M** and **t**, we solve the following equations:

$$\mathbf{d}_{i,j} = \mathbf{f}(\mathbf{v}_j)|_{j=1}^{n}, \quad \sum_{j=1}^{n} \mathbf{w}_j = 0, \quad \sum_{j=1}^{n} \mathbf{w}_j^T \mathbf{v}_j = 0. \tag{2}$$

This system of linear equations is solved using the LU decomposition to obtain the unknown parameters. Using the predefined interpolation function as given in Eq. 1, we calculate the displacement vectors of all vertices to deform the generic model (see Fig. 7 (a)). Note that the connectivity of the mesh is unchanged.

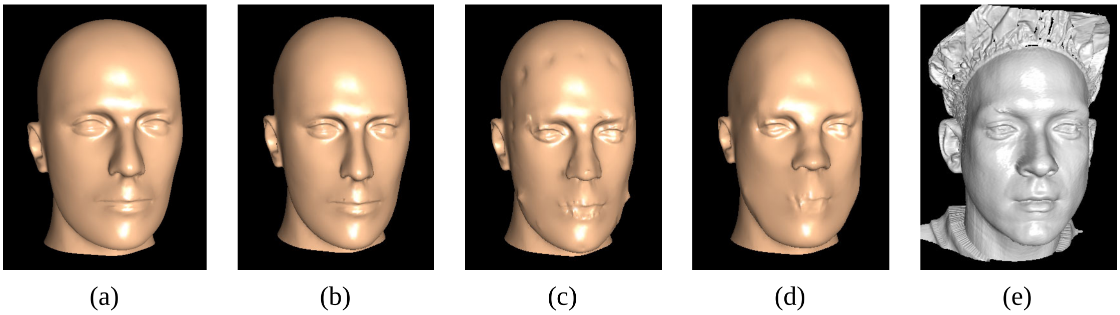

(a) (b) (c) (d) (e)

Figure 6: Comparison of global warping using different basis functions: (a) original generic model; (b) deformation with Gaussian function. (c) with thin-plate spline function; (d) with multi-quadric function; (e) target face scan. Model in (b) is more similar to the original generic model than to the target scanned shape; model in (c) has its local features deformed to approximate the face scan but with undesired bumps; model in (d) has better global and local shape approximation.

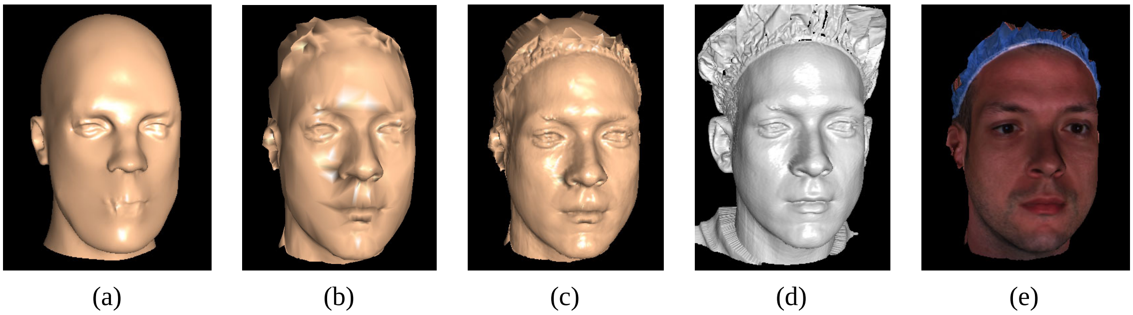

(a) (b) (c) (d) (e)

Figure 7: Model fitting: (a) generic model after global warping; (b) after local deformation; (c) level-2 normal mesh; (d) target scanned model; (e) textured.

5.2 Local Deformation

As shown in Fig. 7 (a), the warping with a small set of correspondences does not produce a perfect surface match. We further improve the shape using a local deformation which ensures that all the generic mesh vertices are truly embedded in the scanned surface. The local deformation is based on the closest points on the surfaces of the generic model and the scanned data. The vertices of the generic model are displaced towards their closest positions on the surface of the scanned data. The polygons of the scanned data are organized into a Binary

Space Partition tree in order to speed up the process of the closest point identification. As each generic mesh vertex samples a scanned point, it takes the texture coordinates of that point for texture mapping. Fig. 7 (b)(e) show the result of local deformation.

5.3 Generating the Multi-Resolution Model

The described processes deform the low-resolution generic model to produce a close approximation to the scanned data, implying surface simplification. In order to accurately represent the high-resolution surface detail, we use a quaternary subdivision scheme to construct the subdivision hierarchy on top of the *base mesh* resulting from the local deformation process. The quaternary subdivision scheme is performed such that for an arbitrary triangle, its three edges are bisected to form three new vertices, and new vertices are then connected to create four sub-triangles which are all similar to the original triangle. The newly generated vertices by subdivision are not necessarily lying on the scanned surface. We reconstruct surface details in a hierarchical manner by taking advantages of the normal mesh representation [37]. A normal mesh starts from the base mesh, and an intermediate level of the normal mesh is defined by the normal offsets from the subdivision mesh of the previous level to the scanned surface (see Figure 8). On each level of the resulting hierarchy, we thus obtain a triangular mesh with vertices interpolating the high-resolution scanned surface. The base meshes of all face scans have the same topology. Hence, the intermediate level normal meshes of all face scans have the same topology. And a spatial correspondence is established hierarchically by the intermediate level normal meshes. In our experiments, normal meshes at level 2 are used to strike a good balance between approximation accuracy and computational cost (see Fig. 7 (c)).

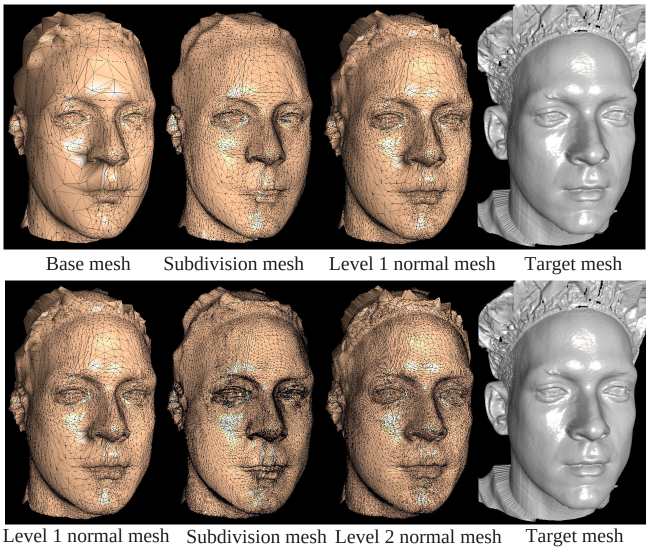


Figure 8: Construction of the normal mesh level by level.

6 Region-Based Face Shape Synthesis

6.1 Forming Local Shape Spaces

To synthesize local shapes of facial features, we firstly form *local shape spaces* using PCA. The model fitting process generates the necessary vertex-to-vertex correspondence across 3D faces in the database, which is the prerequisite of PCA. We have implemented the possibility to select a combination of facial features for face synthesis. Since all face scans are in correspondence through mapping onto the generic model, it is sufficient to define these regions on the generic model. We manually partition the generic mesh into four regions (eyes, nose, mouth and chin), and create a 3D mesh mask to store the subdivision information, where the subregion index for each vertex is recorded. The segmentation is transferred to the multi-resolution normal meshes to generate individualized feature shapes with correspondences (see Fig. 9). Note that we normalize all face scans with respect to the rigid rotation and translation of the face before fitting the generic model. Thus, PCA can be performed directly on the obtained data sets of feature shapes.

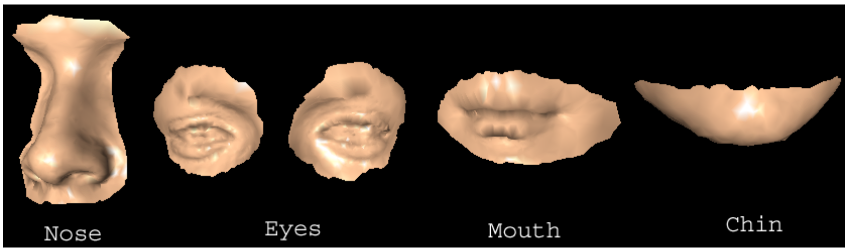


Figure 9: Meshes of four facial features decomposed from the level 2 normal mesh shown in Fig. 8.

Given the set $\{F\}$ of features, we obtain a compact representation for the meshes of each facial feature using PCA. PCA of mesh vertex positions for each feature yields an orthogonal mesh basis, which we term *eigenmeshes*. Computing principal components with the Euclidean norm is equivalent to computing the singular value decomposition (in the case of a square symmetric matrix it is equivalent to eigenanalysis). Let $\{F_i\}_{i=1,\ldots,M}$ be a set of example meshes of feature F, each mesh being associated to one of the M meshes of the database. These meshes are represented as vectors that contain the x, y, z coordinates of the n_F vertices $F_i = (x_{i1}, y_{i1}, z_{i1}, \ldots, x_{in_F}, y_{in_F}, z_{in_F}) \in \mathcal{R}^{3n_F}$. For each feature F we construct a rectangular matrix, $\mathbf{A}_F$, of size $3n_F \times M$, whose columns consist of the x, y, and z components of the vertex positions on the feature region. In the singular value decomposition, $\mathbf{A}_F = \mathbf{U}_F \mathbf{D}_F \mathbf{V}_F^T$, the matrix $\mathbf{U}_F$ has the same size as $\mathbf{A}_F$ and consists of columns of eigenmeshes for feature F in the same block column format that was used to build $\mathbf{A}_F$. The singular values λ_j^F, in the diagonal matrix $\mathbf{D}_F$, identify the importance that each eigenmesh has in reproducing the example meshes (they relate to the proportion of variation explained by each principal component). The matrix $\mathbf{V}_F$ and the singular values combine to give the coordinates of our example meshes in the eigenmesh basis. We denote Φ_j^F the eigenmesh in the basis of feature F with importance j where j goes from 1 (the principal component) up to M.

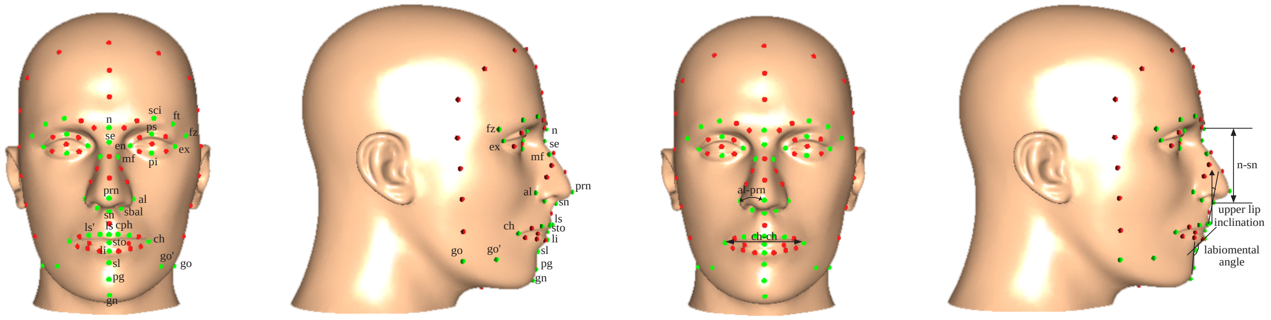
We can truncate each eigenmesh basis expansion knowing that the error will be minimized in the least squares sense [40]. Letting $k_F < M$ be the size of the truncated basis set of feature F, we choose the k_F such that $\sum_{j=1}^{k_F} \lambda_j^F \geq \tau \sum_{j=1}^{M} \lambda_j^F$, where τ defines the proportion of the total variation one wishes to explain (98% for each feature in our experiments). The equation for reconstructing the mesh shape for an arbitrary configuration can then be written as:

$$F(\mathbf{a}) = \Phi_0^F + \sum_{j=1}^{k_F} a_j^F \Phi_j^F \qquad (3)$$

where Φ_0^F is the mean feature shape and a_j^F gives the *coordinates* of the feature shape in terms of the reduced eigenmesh basis. Eq. 3 provides a powerful model for shape synthesis and the model parameter is $\mathbf{a} = \{a_1^F, a_2^F, \ldots, a_{k_F}^F\}$.

6.2 Anthropometric Parameters

Although eigenmeshes represent the most salient directions of the feature shape variation in the dataset, they bear little resemblance to the underlying interdependent structure of biological forms and hardly correspond to the face shape description used in human language. Arguably, face anthropometry provides a set of meaningful measurements or shape parameters that allow the most complete control over the shape of the face. Farkas [31] describes a widely used set of measurements for characterizing the human face. The measurements are taken between the landmark points defined in terms of visually-identifiable or palpable features on the subject's face using carefully specified procedures and measuring instruments. Such measurements use a total of 47 landmark points for describing the face. As described in Section 4, each example in our face scan database is equipped with 86 landmarks. Following the conventions laid out in [31], we have chosen a subset of 38 landmarks from the standard landmark set for anthropometric measurements (see Fig. 10 (a)). This specific set of landmarks used in our system is listed in Appendix A.



(a) (b)

Figure 10: (a) Head geometry with anthropometric landmarks (green dots). The landmark names are taken from [31]. (b) Anthropometric measurements [31].

Landmarks	Measurement Name	Landmarks	Measurement Name
mf-mf	Nasal root width	n-prn	Nasal bridge length
al-al	Nose width	al-prn	Ala surface length
sbal-sbal	Alar base width	al-sn	Alar point-subnasale length
sbal-sn	Nostril floor width	n-prn	Inclination of the nasal bridge
sn-prn	Nasal tip protrusion	sn-prn	Inclination of the columella
en-se	Nasal root depth	al-prn	Inclination of the alar-slope line
en-se	Nasal root slope	n-se-prn	Nasofrontal angle
al-prn	Ala length	al-prn-al	Ala-slope angle
al-mf	Nasal bridge angle	se-prn-sn	Nasal tip angle
n-sn	Nose height	prn-sn-ls	Nasolabial angle

Table 1: Anthropometric measurements of the nose.

Farkas [31] describes a total of 132 measurements on the face and head. They are categorized into five types. As shown in Fig. 10 (b), $ch - ch$ refers to the **shortest distance** between the landmarks at the corners of the mouth, $n - sn$ refers to the **axial distance** between the midpoint of the nasofrontal suture and junction between the lower border of the nasal septum, $al - prn$ refers to the **tangential distance** measured on the face surface from the most lateral point on the nasal ala to the nose tip, the **angle of inclination** is exemplified by the inclination of the upper lip $sn - ls$ with respect to the vertical axis, and the **angle between locations** is exemplified by the labiomental angle (the angle at the lower lip).

In our work, 52 anthropometric measurements related to four facial features are chosen as the shape control parameters. As an example, Table 1 lists the nasal measurements used in our work. There are 7 shortest distance measurements, 3 axial distances, 2 tangential distances, 4 inclination angles and 4 angles between locations. The example models are placed in the standard posture for anthropometric measurements. The measurements are computed using the Euclidean coordinates of landmarks. Particularly, the axial distances correspond to the x, y, and z axes of the world coordinate system. Such a systematic collection of anthropometric measurements are taken through all example models in the database to determine their locations in a multi-dimensional measurement space. These locations are used later to guide the shape synthesis as described in Section 6.3.

6.3 Feature Shape Synthesis

From the previous stage we obtain a set of examples of each feature with measured shape characteristics, each of them consisting of the same set of dimensions, where every dimension is an anthropometric measurement. As each measurement has a different average and variation, it is prohibitive to define the measurement axis by directly using the measurements unit. Instead, the example measurements are normalized. Generally, we assume that an example model F_i of feature F has m_F dimensions, where each dimension is represented by a value in the interval (0,1]. A value of 1 corresponds to the maximum measurement value of the dimension. The measurements of F_i can then be represented by the vector

$$\mathbf{q}_i^F = [q_{i1}^F, \dots, q_{im_F}^F], \forall j \in [1, m_F] : q_{ij}^F \in (0, 1] \tag{4}$$

This is equivalent to projecting each example model F_i into a *measurement space* spanned by the m_F selected anthropometric measurements. The location of each example in the measurement space is $\mathbf{q}_i^F$.

With the input shape control thus parameterized, our goal is to generate a new deformation of the generic mesh by computing the corresponding eigenmesh coordinates with control through the measurement parameter. Given an arbitrary input measurement $\mathbf{q}^F$ in the measurement space, such controlled deformation should interpolate the example models. To do this we interpolate the eigenmesh coordinates of the example models and obtain smooth range over the measurement space. Again, the RBFs are employed in our shape interpolation scheme. Given the input anthropometric control parameters, a novel output model with the desired shapes of facial features is obtained in run time by blending the example models (see Fig. 11). Our scheme first evaluates the predefined RBFs at the input measurement vector and then computes the eigenmesh coordinates by blending those of the example models with respect to the produced RBF values and pre-computed weight values. Finally, the output model with the desired feature shape is generated by evaluating the local shape morphing model (Eq. 3) at those eigenmesh coordinates.

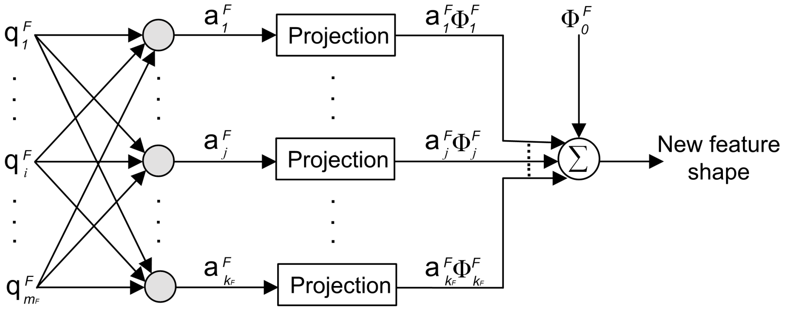


Figure 11: Generating a new facial feature shape by blending example models through interpolation of their eigenmesh coordinates.

The interpolation is multi-dimensional. Consider a $\mathcal{R}^{m_F} \to \mathcal{R}$ mapping, the interpolated eigenmesh coordinates $a_j^F(\cdot) \in \mathcal{R}$, $1 \leq j \leq k_F$ at an input measurement vector $\mathbf{q}^F \in \mathcal{R}^{m_F}$ are computed as:

$$a_j^F(\mathbf{q}^F) = \sum_{i=1}^{M} \gamma_{ij} R_i(\mathbf{q}^F) \quad \text{for} \quad 1 \leq j \leq k_F, \tag{5}$$

where $\gamma_{ij} \in \mathcal{R}$ are the radial coefficients and M is the number of example models. Let $\mathbf{q}_i^F$, $1 \leq i \leq M$ be the measurement vector of an example model. The radial basis function $R_i(\mathbf{q}^F)$ is a multi-quadric function of the Euclidean distance between $\mathbf{q}^F$ and $\mathbf{q}_i^F$ in the measurement space:

$$R_i(\mathbf{q}^F) = \sqrt{\| \mathbf{q}^F - \mathbf{q}_i^F \|^2 + \rho_i^2} \quad \text{for} \quad 1 \leq i \leq M, \tag{6}$$

where ρ_i is the locality parameter used to control the behavior of the basis function and determined as the Euclidean distance between $\mathbf{q}_i^F$ and the closest other example measurement vector.

The j-th eigenmesh coordinate of the i-th example model, $a_j^{F_i}$ corresponds to the measurement vector of the i-th example model, $\mathbf{q}_i^F$. Eq. 5 should be satisfied for $\mathbf{q}_i^F$ and $a_j^{F_i}$ ($1 \leq i \leq M$):

$$a_j^{F_i}(\mathbf{q}_i^F) = \sum_{i=1}^{M} \gamma_{ij} R_i(\mathbf{q}_i^F) \quad \text{for} \quad 1 \leq j \leq k_F \tag{7}$$

Eq. 7 can be expressed in the matrix form

$$\mathbf{RY} = \mathbf{A} \tag{8}$$

where $\mathbf{Y} \in \mathcal{R}^{M \times k_F}$ is the matrix of the unknown radial coefficients γ_{ij}, and $\mathbf{R} \in \mathcal{R}^{M \times M}$ and $\mathbf{A} \in \mathcal{R}^{M \times k_F}$ are the matrices defined by the radial bases and the eigenmesh coordinate of the example models, respectively, such that $\mathbf{R}_{ij} = R_j(\mathbf{q}_i^F)$ and $\mathbf{A}_{ij} = a_j^{F_i}$. γ_{ij} are obtained by solving Eq. 8 using the LU decomposition. We can then generate the eigenmesh coordinates corresponding to the input measurement vector $\mathbf{q}^F$ according to Eq. 5.

After the shape interpolation procedure, the surrounding facial areas should be blended with the deformed features to generate a seamlessly smooth face mesh. We apply a gradual geometric blending along the subregion boundaries (Appendix B). Fig. 12 shows the effect of this scheme employed in feature shape synthesis.

Our system provides two ways to measurement generation: 1) interactive user specification and 2) automatically random generation. The first module allows the user to produce a face with needed specific characteristics using descriptions. The measurement values are chosen within [0,1]. All user operations result in an immediate update of the displayed face. To facilitate user interaction, we have implemented an operation history with multiple levels of “undo”. The random generation module allows the construction of a random face, in case the user needs a representative face from the population and does not care about specific appearance. It is assumed that the measurements follow the Gaussian normal distribution [31]. Therefore, random value for each input measurement is generated by sampling the distribution with its mean and variance. Given the shape synthesis framework we use, a feature shape can be generated to fit the generated suite of control parameters.

Our system supports exaggeration or caricature. Psychological findings [21] have shown that making faces more typical by caricaturing techniques increases subjects’ ability to recognize faces. Caricaturing is achieved by morphing a feature away from the average. In our framework, the output eigenmesh coordinates are multiplied by a positive scalar factor η to create a caricatured facial feature. η is adjustable by the user for different levels of caricature, although its default value is set to one.

7 Region-Based Face Texture Synthesis

To synthesize textures of facial features, we form *local texture spaces* by using PCA. The variations of face textures in the database depend exclusively on the intrinsic parameters - face shape and color/texture. Applying PCA to a set of face images requires normalization to remove texture variation due to shape difference, and correspondences must be found between face images. In our case, however, texture correspondences are implicit in the texture coordinates of the two associated face meshes. Since every face generated from one generic model has a similar characteristic for texture coordinates, we can produce shape-free face textures by constructing a parameterization of the 3D generic mesh over a 2D image plane.

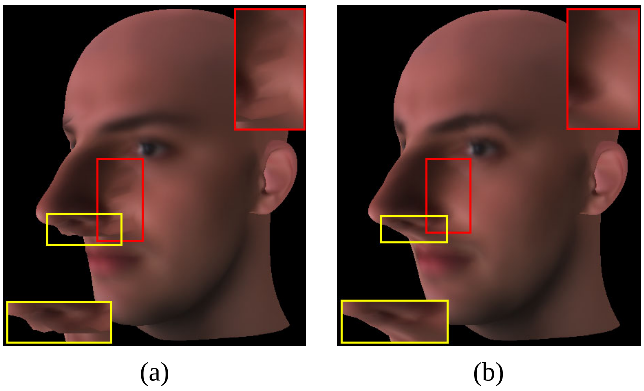


Figure 12: Synthesis of the nose shape: (a) Without shape blending, the obvious geometric discontinuities around boundary of the nose region impair realism of the morphing to a large extent. (b) Using our approach, geometries of the feature region and surrounding area are smoothly blended around their boundary.

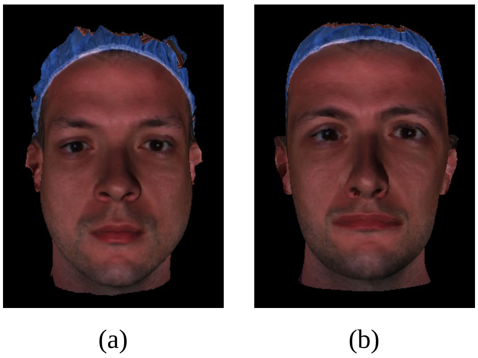


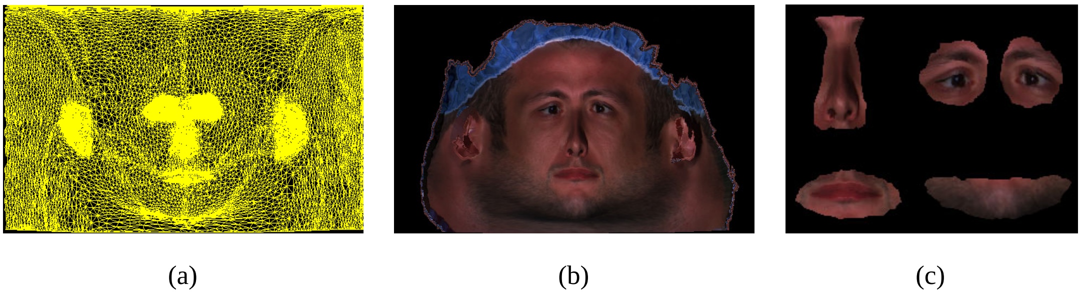
Figure 13: (a) A base mesh with texture mapping. (b) Texture transferred to the original undeformed generic model.

Given the vertex-wise correspondence between a fitted generic mesh (base mesh) and original undeformed generic mesh, it is trivial to transfer a texture map between them. Each vertex on the original generic mesh simply takes the texture coordinates of its corresponding vertex on the fitted mesh for texture mapping (see Fig. 13). We want to parameterize the 3D generic head mesh over a 2D domain [0, 1]^2. In our case, the head mesh is topologically equivalent to a part of a plane. Thus we can “flatten” the generic head mesh to a part of a 2D image plane that is bounded by its boundary curve around the neck. We implement a cylindrical projection to obtain a cylindrical face mesh (see Fig. 14 (a)). Each vertex of the 2D cylindrical mesh has cylindrical coordinates with corresponding longitude (0-360 degrees) along the x-axis and vertical height along the y-axis. The mapping of world coordinates $\mathbf{x} = (x, y, z)$ to 2D cylindrical coordinates (u, v) uses

$$u = \tan^{-1}(\frac{x}{z}), \qquad v = y \tag{9}$$

The resulting (u, v) coordinates map to a suitable aspect and resolution image (800×505 in our experiment). We also map the original generic mesh rendered with transferred texture to the image plane using the same cylindrical projection. The result is a 800×505 *cylindrical texture image* in which each pixel value represents the surface color of the texture-mapped face surface in cylindrical coordinates (see Fig. 14 (b)).

Since the generic mesh has been partitioned into four feature regions, the cylindrical texture image can be



(a) (b) (c)

Figure 14: (a) Texture mesh parameterization. (b) Cylindrical texture image. (c) Segmented textures of four facial features.

Attribute description	Attribute description
Dark skin color	Darkness of the mustache below lower lip
White skin color	Dark lip color
Yellowish skin color	Natural red lip color
Reddish skin color	Cosmetic red lip color
Darkness of the mustache over upper lip	

Table 2: Texture attributes for mouth.

divided into corresponding patches (see Fig. 14 (c)). As with the shape synthesis, a texture PCA space is constructed for a data set of M exemplar local textures of each feature. For each facial feature, we define a set of distinct texture attributes to build a *texture attribute space* where each attribute represents a separate axis in this space. As an example, Table 2 lists the texture attributes for mouth. We manually assign the attribute values (in the interval (0,1]) that describe the markedness of the attributes to each example texture, parameterizing the texture in the texture attribute space. Given a new set of input texture attribute values, the desired characteristics can be synthesized on the mean or an individual cylindrical full-head texture image[1] (see Fig. 15) by blending the example local textures through RBF-based interpolation, as described in Section 6.3. To avoid the image discontinuity, we do a gradual blending along the subregion boundaries (Appendix C).

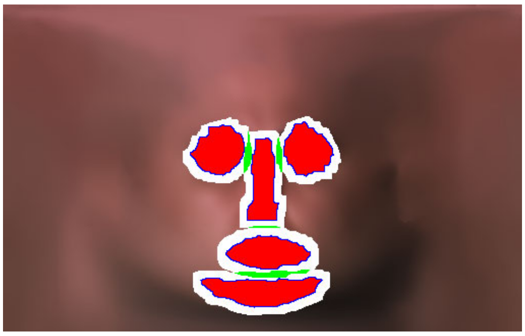
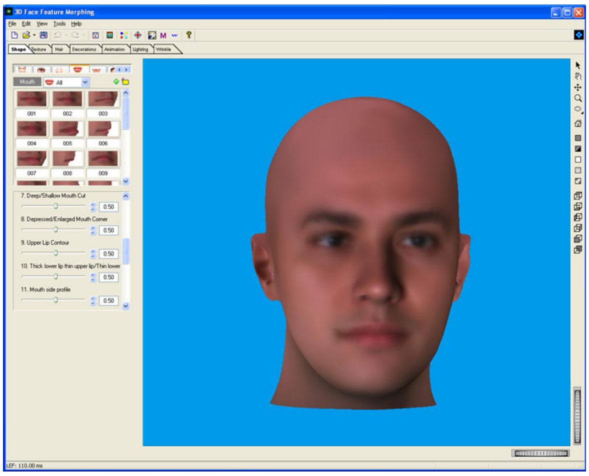


Figure 15: Facial feature regions and blending regions in the mean cylindrical full-head texture image: the local feature regions are in red with their boundaries in blue. The white areas are the texture blending regions. The overlap of two blending regions is in green.

8 Results

Our method has been implemented in an interactive system with C++/OpenGL. A GUI snapshot is shown in Fig. 16. Upon start, an average head is shown in the display area of the GUI. The average head is computed as

[1] The reader is referred to our paper [90] for detailed description of the full-head texture generation technique.


3D Face Feature Morphing
File Edit View Tools Help
Shape Texture Hair Decorations Animation Lighting Wrinkle
Mouth All
001 002 003
004 005 006
007 008 009

7. Deep/Shallow Mouth Cut 0.50
8. Depressed/Enlarged Mouth Corner 0.50
9. Upper Lip Contour 0.50
10. Thick lower lip thin upper lip/Thin lower 0.50
11. Mouth side profile 0.50

Figure 16: GUI of our system.

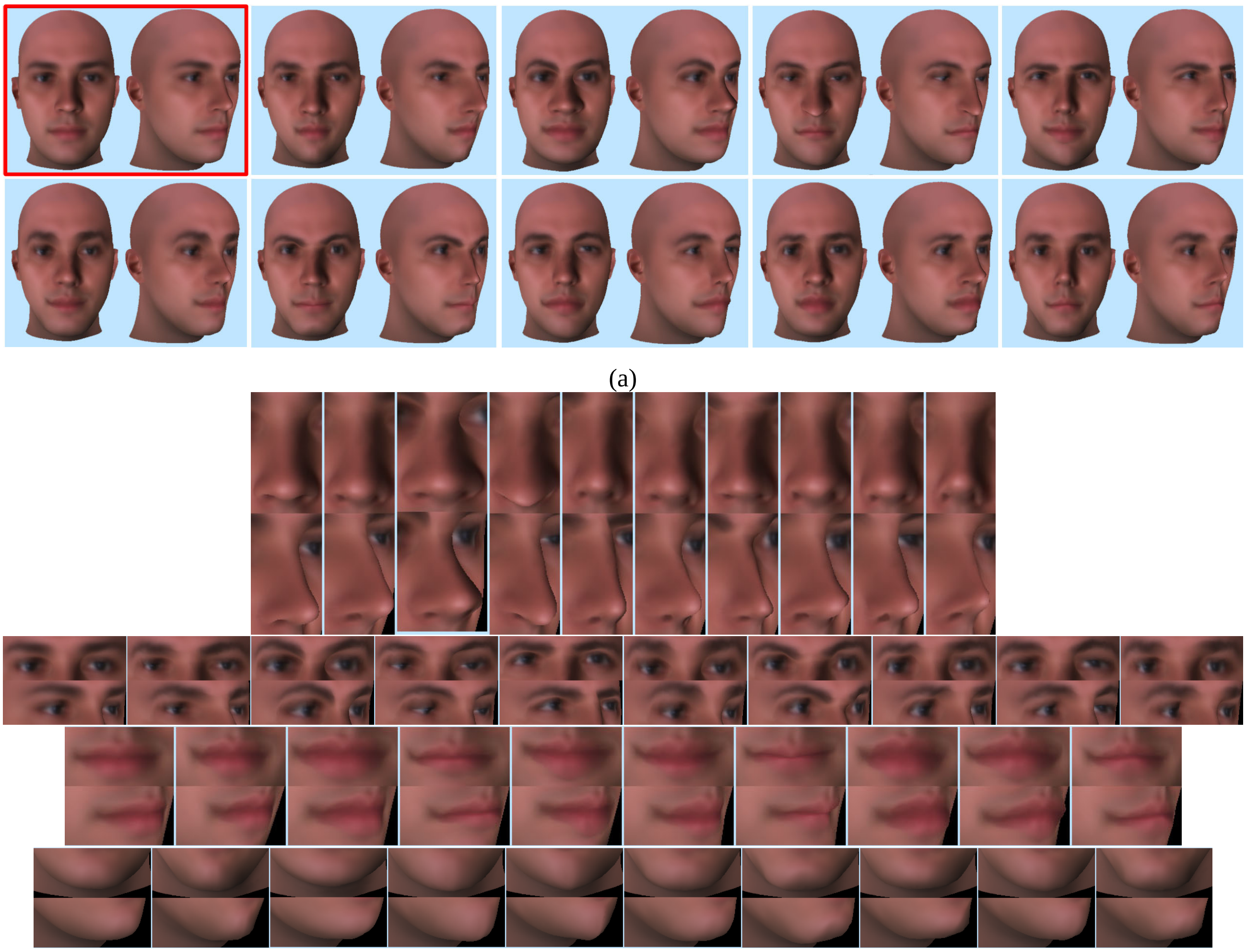
	Shape				Texture			
	Eyes	Nose	Mouth	Chin	Eyes	Nose	Mouth	Chin
Numer of eigen modes used for synthesis	23	26	20	18	32	21	26	17
Number of control parameters	13	20	12	7	10	5	9	6

Table 3: Number of eigen modes and control parameters used in our system.

the average of level-2 normal meshes of 186 example models and is textured with the mean cylindrical full-head texture image. In addition to starting with the average head, the user may also select the desired head model of a specific person from the example database for further editing. Facial features can be modified on a continuous scale using sliders.

Table 3 shows the number of eigenmeshes/eigentextures used for feature synthesis and the number of shape/texture control parameters. Besides modifying facial features using the shape/texture control parameters, our system also allows the user to select the desired feature(s) from a database of pre-constructed typical features, which are shown in the small icons on the upper-left of the GUI. Upon selecting a feature from the database, the feature will be imported seamlessly into the displayed head model and can be further edited if needed. The slider positions for each of the available feature in the database are stored by the system so that their configuration can be restored whenever the feature is chosen for operation. As with local shape/texture editing, the import of features from the database can be restricted to the shape and/or texture of the feature. Such a feature importing mode enables coarse-to-fine modification of features, making the face synthesis process less tedious.

Fig. 17 illustrates a number of synthesized face shapes. A close view of each feature reveals a wide range of variations across the generated faces; for example, clear differences are found in the width of the nose alar wings, the straightness of the nose bridge, the inclination of the nose tip, the roundness of eyes, the distance between eyebrows and eyes, the thickness of mouth lips, the shape of the lip line, the sharpness of the chin, etc. A dynamic morph can be generated by varying the shape parameters continuously between two values, as shown in Fig. 18. Fig. 19 and 20 show some examples of new facial appearances generated by local texture



(a)

(b)

Figure 17: (a) Automatically generated face models by synthesizing the shape of four facial features on the average model (outlined) according to the input anthropometric parameters. (b) Close view of synthesized shapes of individual features.

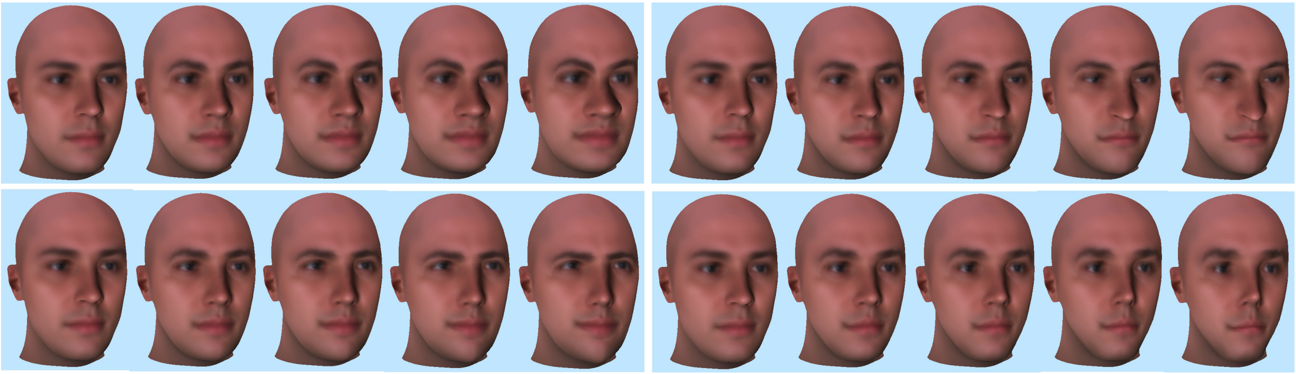


Figure 18: Region-based face shape synthesis (left to right in each example). Face texture is kept unchanged.

synthesis and the snapshots of dynamic feature texture morphing, respectively. The nine generated faces in Fig. 19 exhibit the expected range of texture variation and represent different individuals, encompassing a range of ages and genders. We can also start with any face model of a specific person and edit various aspects of its shape and texture, as shown in Fig. 21.

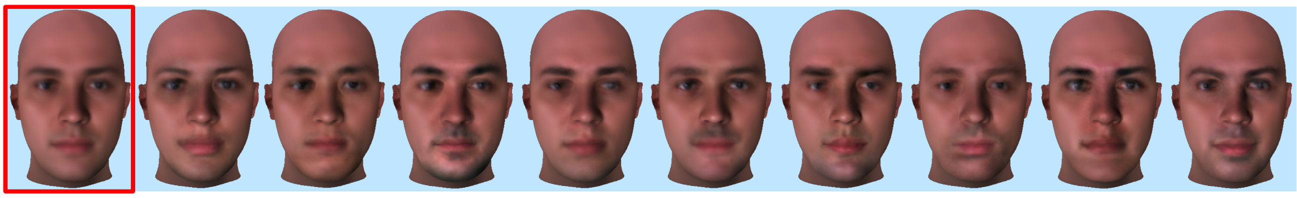


Figure 19: Some novel facial appearances generated by synthesizing the textures of four facial features on the average model rendered with the mean texture (outlined).

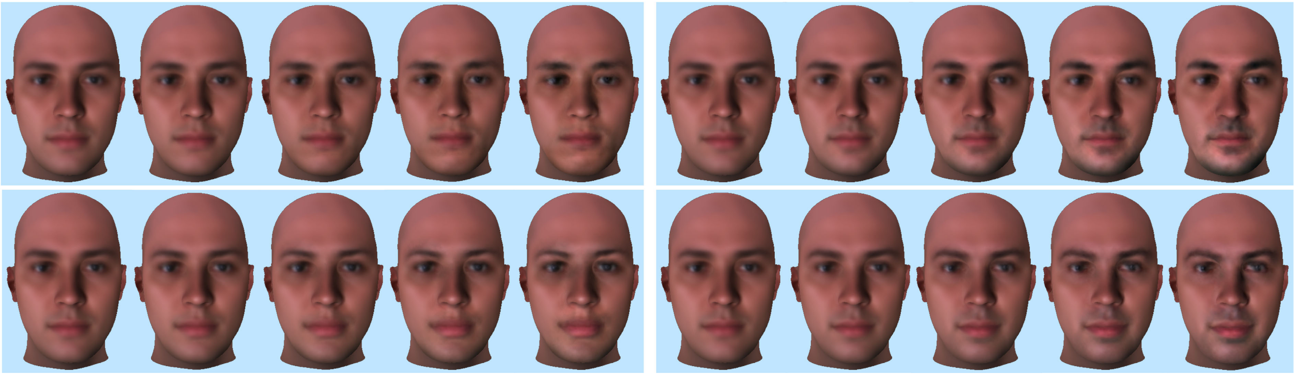


Figure 20: Region-based face texture synthesis (left to right in each example). Face shape is kept unchanged.

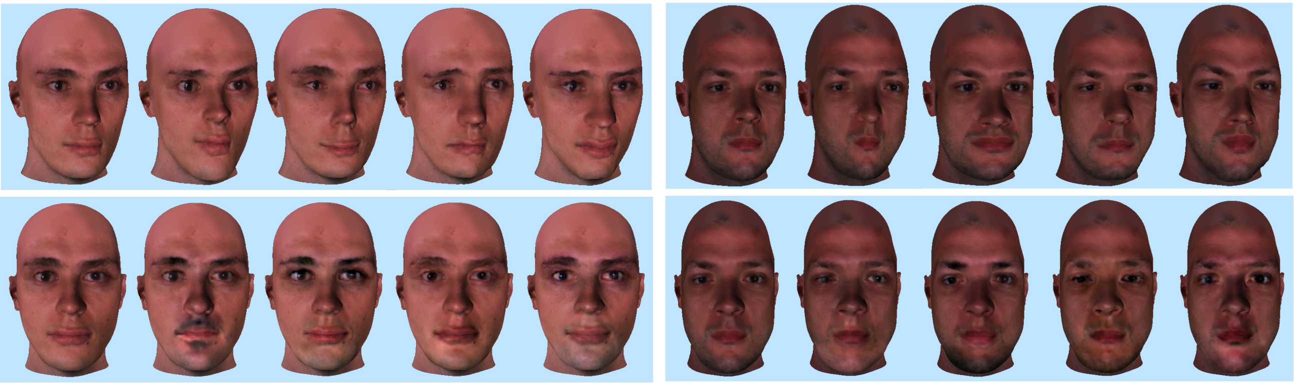


Figure 21: Variation of facial features of two individual faces. Top row: region-based shape synthesis. Bottom row: region-based texture synthesis. The original model is shown in the first image of each example.

In creating virtual characters, sometimes it is desirable to adjust the face so that it has certain features similar to those of a specific person. Therefore, it is useful to be able to transfer desired facial feature(s) between different human subjects. Facial feature transfer falls neatly into our local feature control framework. Our system allows the user to alter a feature of the source model by coercing its shape and texture control measurements to match those of the target feature of an example face. Fig. 22 shows the result of facial feature transfer. Typical features of different example faces can also be transferred in conjunction with each other to the same source model. Fig. 23 shows a composite face built from features of four individuals.

In order to quantify the performance, we arbitrarily selected ten examples in the database for the cross-validation. Each example has been excluded from the example database in training the face synthesis system and its shape and texture measurements were used as a test input to the system. The output model was then compared with the input model. We assess the reconstruction by measuring the maximum, mean, and root

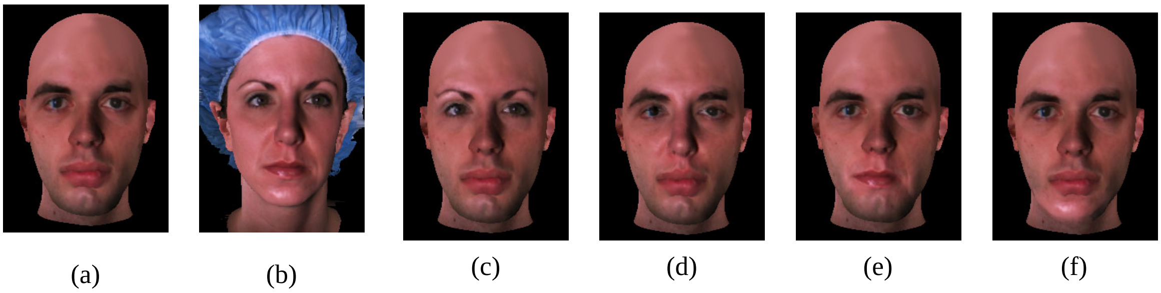


Figure 22: Transfer of facial features. We start with a source model (a), and synthesize facial features of the eyes (c), nose (d), mouth (e) and chin (f) on it by coercing shape and texture control parameters to match those of the target faces (b).

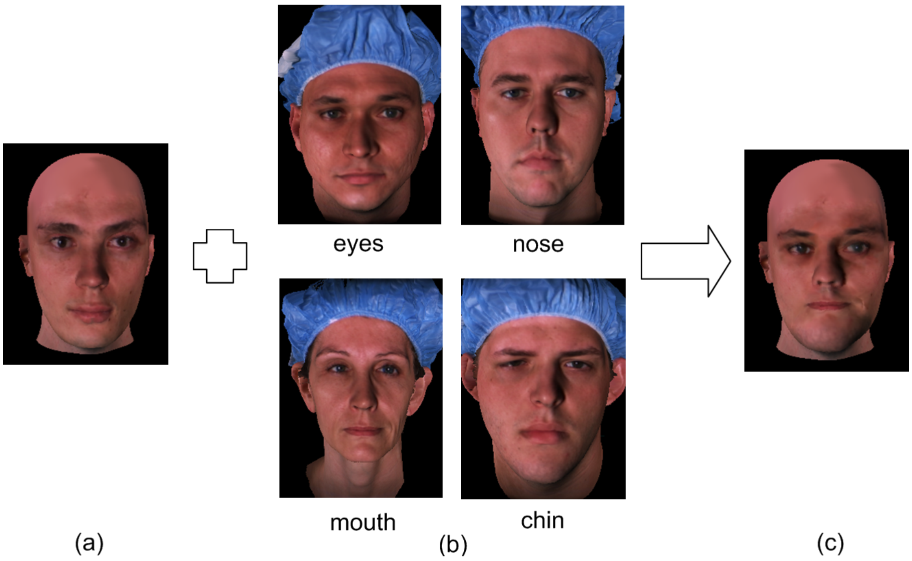


Figure 23: Facial features from four target faces (b) in our database are transferred to the source model (a) to generate a novel composite face (c).

mean square (RMS) errors from the feature regions of the output model to those of the input model. For the feature shapes, errors are computed by the distances between the corresponding vertex positions. Table 4 shows the average errors measured for the ten reconstructed models. The errors are given using both absolute measures (/mm) and as a percentage of the diameter of the output head model bounding box. For the feature textures, errors are measured as the differences between colors of the corresponding pixels of the input and output cylindrical full-head textures in terms of Euclidean distance in the RGB color space. The average errors in absolute measurements (the color value of each channel is in the interval [0,255]) are given in Table 4.

	Shape				Texture			
	Eyes	Nose	Mouth	Chin	Eyes	Nose	Mouth	Chin
Average max.	3.85 (0.91%)	3.55 (0.84%)	6.58 (1.65%)	4.46 (1.06%)	18.3	15.9	23.5	25.7
Average mean	1.37 (0.33%)	1.62 (0.38%)	2.04 (0.49%)	2.57 (0.57%)	7.8	7.2	11.7	9.4
Average RMS	1.93 (0.46%)	2.23 (0.53%)	2.84 (0.67%)	3.62 (0.86%)	10.6	10.1	15.8	13.1

Table 4: Cross validation results of our 3D face synthesis system.

Table 5 shows the time cost of different procedures in the current unoptimized implementation of our system

Process	T_I	T_A	
Offline processing			
Shape			
Feature point identification	3-5 min	6 sec	×186
Global warping	N/A	2 sec	
Local deformation	N/A	0.8 sec	
Multi-resolution model generation	N/A	5 sec	
Computing the eigenmeshes by PCA	N/A	2 hrs	
Computing the eigenmesh coordinates	N/A	0.5 sec	
Computing the anthropometric measurements	N/A	0.2 sec	
LU decomposition	N/A	2 min	
Texture			
Texture attribute value assignment	10-12 min	N/A	×186
Mesh parameterization	N/A	50 ms	
Computing the eigentextures by PCA	N/A	3 hrs	
Computing the eigentexture coordinates	N/A	0.8 sec	
LU decomposition	N/A	2 min	
Runtime			
Feature shape synthesis	N/A	0.6 sec	
Feature texture synthesis	N/A	1 sec	

Table 5: Time consumed for different processes of system implementation. For some processes, the time spent per example is summed up to get the total time consumed for all 186 examples. Notation: time consumed in interactive operation (T_I), time consumed in automatic computation (T_A).

on a 2.8 GHz PC with 1GB RAM. Most time is consumed in the offline processing stage. Even though the offline processing steps take up considerable time, this does not impair usability to a large extent due to the automation (beyond initial feature point identification and assignment of texture attribute values) and their one-time computation nature. At runtime, our scheme spends about one second in generating a new face using the refinement level-2 head model (36,384 triangles) rendered with a 800×505 texture image upon receiving the input parameters.

9 Conclusion and Future Work

We have presented an automatic runtime system for generating varied realistic geometric models of human faces based on synthesis of facial features according to intuitive control parameters. We make use of the scanned face data, which arguably provides the best available resource to regulate the naturalness of modeled faces. A three-step model fitting approach is used for the 3D registration problem, where a generic head model is fitted to each example in a global-to-local fashion. We transform the obtained data sets of facial feature shapes into vector space representations by applying the PCA, and parameterize the feature shape examples using a set of anthropometric measurements. Using the PCA coefficients as a compact shape representation, we approach the feature shape synthesis problem by forming scattered data interpolation functions that are devoted to the generation of desired shape by taking the anthropometric parameters as input. For synthesizing local feature textures, we automatically parameterize the 3D generic head mesh over the 2D image domain to establish the mapping correspondence between example textures. Given the input texture attribute parameters, the feature textures are synthesized at an interactive rate by evaluating the predefined interpolation functions. The resulting system is intuitive to control and fine-grained.

With the existing framework, several areas of improvement may lead to more plausible facial synthesis. We would like to extend our current database to incorporate more 3D face examples of Mongolian and Negroid races as well as to increase diversity of age by including children and more elderly people. Even without collecting more 3D scans, our system can be extended easily to cover a wider range of faces. Given a set of single photographs of additional faces, 3D face models can be reconstructed by fitting a morphable face model [6]. These 3D faces are then included in the database. Even though the reconstructed shapes are in the linear span of the original database, we can map the color values from the photographs to the model by using illumination-correction texture extraction [6], adding new dimensions to the texture vector space.

We would like to increase the number of facial features to choose from. The possible candidates are the cheeks, forehead, upper jaw, and ears. Some facial features (e.g., nose) can be represented by a hierarchy of segments. In the feature synthesis, the user may select segments from any hierarchy level of any facial feature to achieve different levels of editing. In addition, we plan to use some extensions of ASM to achieve more robust feature detection. An automatic determination of the texture attribute values is also one of the future challenges.

In the current system, the effect of editing operations is restricted to the shape and/or texture of a single facial feature. It might happen in some cases (e.g., sketch of the face of a suspect) that the user does not know what detail of all features should be changed. To this end, we would like to improve the shape control by using the facial proportion statistics [32] to model the correlations between measurements of different features. This will allow features to be edited while related features systematically changed to preserve natural proportions.

A limitation of our system is that the current linear PCA model cannot be directly applied to synthesizing high-resolution face details. It would be difficult or impossible to find exact correspondences between high resolution details of example faces. Without correct correspondence, the weighted linear blending blurs small facial features such as wrinkles and pores, making the output model smooth. Synthesis of detailed facial geometry and texture is a difficult problem. One possibility is to use the statistical synthesis technique as in [36] to complement our method, adding wrinkles and other fine-scale face details to the model.

For animating the face, we plan to utilize an anatomy-based model with the skin-muscle-skull structure developed in our previous work [92] as the generic model. By applying local shape morphing techniques to the face surface and the interior skull simultaneously, the whole face structure is adapted. As the anatomy-based model has been designed to produce real-time animation based on simulation of facial muscles and elastic skin properties, the textured head models with the adapted underlying anatomical structure can be animated instantly. Finally, it would be very helpful to use 3D models of hair to cover textured heads with various hair styles. Adding details such as birth marks, scars, or jewelry will be another extension.

References

[1] T. Akimoto, Y. Suenaga, and R. S. Wallace. “Automatic creation of 3D facial models.” *IEEE Computer Graphics and Application*, 13(5):16-22, September 1993.

[2] B. Allen, B. Curless, and Z. Popovic. “The space of human body shapes: Reconstruction and parameterization from range scans.” *Proc. SIGGRAPH’03*, pp. 587-594, 2003.

[3] B. Allen, B. Curless, and Z. Popovic. “Articulated body deformation from range scan data.” *Proc. SIGGRAPH’02*, pp. 612-619, 2002.

[4] Autodesk Maya, http://www.autodesk.com/maya

[5] E. Baker and M. Seltzer. “The mug-shot search problem.” *Proc. Vision Interface’98*, pp. 65-72, 1998.

[6] V. Blanz and T. Vetter. “A morphable model for the synthesis of 3D faces.” *Proc. SIGGRAPH’99*, pp. 187-194, August 1999.

[7] V. Blanz, C. Basso, T. Poggio, and T. Vetter. “Reanimating faces in images and video.” *Proc. EUROGRAPHICS’03*, pp. 641-650, 2003.

[8] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel. “Exchanging faces in images.” *Proc. EUROGRAPHICS’04*, pp. 669-676, 2004.

[9] M. Brand. “Voice puppetry.” *Proc. SIGGRAPH’99*, pp. 21-28, August 1999.

[10] R. Brunelli and O. Mich. “SpotIt! an interactive identikit system.” *Graphical Models and Image Processing*, 58(5): 399-404, 1996.

[11] M. Byun and N. I. Badler. “FacEMOTE: Qualitative parametric modifiers for facial animations.” *Proc. ACM SIGGRAPH Symp. on Comput. Anim.*, pp. 65-71, 2002.

[12] Y. Cao, P. Faloutsos, E. Kohler, and F. Pighin. “Real-time speech motion synthesis from recorded motions.” *Proc. ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* , pp. 347-355, 2004.

[13] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B.C. Mccallum, and T. R. Evans. “Reconstruction and representation of 3d objects with radial basis functions.” *Proc. SIGGRAPH’01*, pp. 67-76, August 2001.

[14] J. Cassell, H. H. Vilhjalmsson, and T. Bickmore. “BEAT: the behavior expression animation toolkit.” *Proc. SIGGRAPH’01*, pp. 477-486, August 2001.

[15] J. Chai, J. Xiao, and J. Hodgins. “Vision-based control of 3D facial animation.” *Proc. ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* , pp. 193-206, 2003.

[16] T.-P. G. Chen, S. Fels. “Exploring gradient-based face navigation interfaces.” *Proc. Graphics Interface’04*, pp. 65-72, 2004.

[17] B. Choe, H. Lee, and H.-S. Ko. “Performance-driven muscle-based facial animation.” *Journal of Visualization and Computer Animation*, 12: 67-79, 2001.

[18] E. Chuang and C. Bregler. “Moodswings: Expressive speech animation.” *Proc. SIGGRAPH’05*, pp. 331-347, 2005.

[19] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. "Active shape models: Their training and applications." *Computer Vision and Image Understanding,* 61(1):38-59, 1995.

[20] D. DeCarlo, D. Metaxas, and M. Stone. "An anthropometric face model using variational techniques." *Proc. SIGGRAPH'98,* pp. 67-74, July 1998.

[21] K. A. Deffenbacher, J. Johanson, T. Vetter, and A. J. O'toole. "The face typicality-recognizability relationship: encoding or retrieval locus?" *Memory and Cognition,* 28(7):1173-1182, 2000.

[22] Z. Deng, U. Neumann, J. P. Lewis, T. Y. Kim, M. Bulut, and S. Narayanan. "Expressive facial animation synthesis by learning speech co-articulations and expression spaces." *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1523-1534, 2006.

[23] Z. Deng and U. Neumann. "eFASE: Expressive facial animation synthesis editing with phoneme-isomap controls." *Proc. ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* , pp. 251-260, 2006.

[24] Z. Deng, P. Y. Chiang, P. Fox, and U. Neumann. "Animating blendshape faces by cross-mapping motion capture data." *Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics,* pp. 43-48, 2006.

[25] Z. Deng, J. P. Lewis, and U. Neumann. "Automated eye motion using texture synthesis." *IEEE Computer Graphics and Application*, 25(2):24-30, March/April 2005.

[26] S. DiPaola. "Extending the range of facial types." *Journal of Visualization and Computer Animation,* 2(4): 129-131, 1991.

[27] E-FIT^{TM} from Aspley Ltd., http://www.efit.co.uk

[28] T. Ezzat, G. Geiger, and T. Poggio. "Trainable videorealistic speech animation." *Proc. SIGGRAPH'02,* pp. 388-398, July 2002.

[29] I. Essa and S. Basu. "Modeling, tracking and interactive animation of facial expressions and head movements using input from video." *Proc. Computer Animation'96,* pp. 68-79, June 1996.

[30] FaceGen Modeller 3.0 from Singular Inversions Inc., http://www.FaceGen.com

[31] L. G. Farkas. *Anthropometry of the Head and Face,* Raven Press, 1994.

[32] L. G. Farkas. *Anthropometric Facial Proportions in Medicine,* Thomas Books, 1987.

[33] D. Fidaleo, J-Y. Noh, T. Kim, R. Enciso and U. Neumann. "Classification and volume morphing for performance-driven facial animation." *International Workshop on Digital and Computational Video,* 2000.

[34] P. Fua and C. Miccio. "Animated heads from ordinary images: A least-squares approach." *Computer Vision and Image Understanding,* 75(3):247-259, 1999.

[35] B. Guenter, C. Grimm, D. Wood, H, Malvar, and F. Pighin. "Making faces." *Proc. SIGGRAPH'98,* pp. 55-66, 1998.

[36] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. "A statistical model for synthesis of detailed facial geometry." *Proc. SIGGRAPH'06,* pp. 1025-1034, August 2006.

[37] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. "Normal meshes." *Proc. SIGGRAPH'00,* pp. 95-102, 2000.

[38] Identi-Kit.NET^{TM} from Smith & Wesson, https://www.identikit.net

[39] H. H. S. Ip and L. Yin. "Constructing a 3D individualized head model from two orthogonal views." *The Visual Computer,* 12:254-266, 1996.

[40] I. T. Jollife. *Principal Component Analysis.* Springer Verlag, New York, 1986.

[41] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. "Head shop: Generating animated head models with anatomical structure." *Proc. ACM SIGGRAPH Symp. on Comput. Anim.,* pp. 55-64, 2002.

[42] K. Kähler, J. Haber, and H.-P. Seidel. "Geometry-based muscle modeling for facial animation." *Proc. Graphics Interface'01,* pp. 37-46, June 2001.

[43] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. "Simulation of facial muscle actions based on rational free form deformatios." *Proc. EUROGRAPHICS'92,* pp. 59-69, 1992.

[44] E. Keeve, S. Girod, P. Pfeifle, and B. Girod. "Anatomy-based facial tissue modelling using the finite element method." *Proc. IEEE Visualization'96,* pp. 21-28, 1996.

[45] S. A. King and R. E. Parent. "Creating speech-synchronized animation." *IEEE Transactions on Visualization and Computer Graphics*, 11(3):341-352, 2005.

[46] R. Koch, M. Gross, F. Carls, D. Buren, G. Fankhauser, and Y. Parish. "Simulating facial surgery using finite element models." *Proc. SIGGRAPH'96,* pp. 421-428, August 1996.

[47] R. Koch, M. Gross, and A. Bosshard. "Emotion editing using finite elements."*Proc. EUROGRAPHICS'98,* pp. 295-302, 1998.

[48] S. Kshirsagar and N. M. Thalmann. "Visyllable based speech animation."*Proc. EUROGRAPHICS'03,* pp. 632-640, 2003.

[49] C. Kuo, R. Huang, and T. Lin. "3-D facial model estimation from single front-view facial image." *IEEE Trans. on Circuits and Systems for Video Technology,* 12(3):183-192, March 2002.

[50] S.-H. Lee and D. Terzopoulos. "Heads up!: biomechanical modeling and neuromuscular control of the neck." *Proc. SIGGRAPH'06,* pp. 1188-1198, August 2006.

[51] S. P. Lee, J. B. Badler, and N. I. Badler. "Eyes alive." *Proc. SIGGRAPH'02,* pp. 637-644, July 2002.

[52] W. S. Lee and N. Magnenat-Thalmann. "Fast head modeling for animation." *Journal Image and Vision Computing,* 18(4): 355-364, March 2000.

[53] Y. Lee, D. Terzopoulos, and K. Waters. "Realistic modeling for facial animation." *Proc. SIGGRAPH'95,* pp. 55-62, August 1995.

[54] J. P. Lewis. "Algorithms for solid noise synthesis." *Proc. SIGGRAPH'89,* pp. 263-270, 1989.

[55] J. Lewis, M. Cordner, and N. Fong. "Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation." *Proc. SIGGRAPH'00,* pp. 165-172, July 2000.

[56] H. Li, P. Roivainen, and R. Forchheimer. "3-D motion estimation in model-based facial image coding." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6): 545-555, 1993.

[57] Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen. "Rapid modeling of animated faces from video." *Journal of Visualization and Computer Animation*, 12(4): 227-240, September 2001.

[58] J. Ma, R. Cole, B. Pellom, W. Ward, and B. Wise. "Accurate visible speech synthesis based on concatenating variable length motion capture data." *IEEE Transactions on Visualization and Computer Graphics*, 12(2):266-276, 2006.

[59] N. Magnenat-Thalmann, H. Minh, M. deAngelis, and D. Thalmann. "Design, transformation and animation of human faces." *The Visual Computer,* 5:32-39, 1989.

[60] K. Na and M. Jung. "Hierarchical retargetting of fine facial motions."*Proc. EUROGRAPHICS'04,* pp. 687-695, 2004.

[61] M. Nahas, H. Huitric, and M. Saintourens. "Animation of a B-spline figure." *The Visual Computer,* 3(5):272-276, 1988.

[62] J. Y. Noh and U. Neumann. "Expression cloning." *Proc. SIGGRAPH'01,* pp. 277-288, August 2001.

[63] J. Y. Noh and U. Neumann. *A survey of facial modeling and animation techniques.* USC Technical Report 99-705, USC, Los Angeles, CA, 1999.

[64] I. K. Park, H. Zhang,V. Vezhnevets, and H. K. Choh. "Image-based photorealistic 3D face modeling." *Proc. IEEE Automatic Face and Gesture Recognition,* pp. 49-54, 2004.

[65] F. I. Parke and K. Waters. *Computer Facial Animation.* AK Peters, Wellesley, MA, 1996.

[66] F. I. Parke. "Parameterized models for facial animation." *IEEE Computer Graphics and Application,* 2(9): 61-68, November 1982.

[67] F. I. Parke. *Computer generated animation of faces.* Master's thesis, University of Utah, Salt Lake City, 1972.

[68] M. Patel and P. Willis. "FACES: the facial animation, construction and editing system." *Eurographics'91,* pp. 33-45, 1991.

[69] S. Platt and N. Badler. "Animating facial expressions." *Proc. SIGGRAPH'81,* pp. 245-252, 1981.

[70] F. Pighin, R. Szeliski, and D. H. Salesin. "Resynthesizing facial animation through 3d model-based tracking." *Proc. ICCV'99,* pp. 143-150, 1999.

[71] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. "Synthesizing realistic facial expressions from photographs." *Proc. SIGGRAPH'98,* pp. 75-84, July 1998.

[72] PROfit^{TM} from ABM United Kingdom Ltd., http://www.abm-uk.com/uk/products/profit.asp

[73] H. Pyun, Y. Kim, W. Chae, H. W. Kang, and S. Y. Shin. "An example-based approach for facial expression cloning." *Proc. ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* , pp. 167-176, 2003.

[74] C. Rose, M. Cohen, and B. Bodenheimer. "Verbs and adverbs: Multidimensional motion interpolation using RBF." *IEEE Computer Graphics and Application,* 18(5):32-40, 1998.

[75] H. Seo and N. Magnenat-Thalmann. "Automatic modeling of human bodies from sizing parameters." *Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics*, pp. 19-26, 2003.

[76] E. Sifakis, A. Selle, A. Robinson-Mosher, and R. Fedkiw . "Simulating speech with a physics-based facial muscle model." *Proc. ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* , pp. 261-270, 2006.

[77] E. Sifakis, I. Neverov, and R. Fedkiw. “Automatic determination of facial muscle activations from sparse motion capture marker data.” *Proc. SIGGRAPH’05,* pp. 417-425, 2005.

[78] P.-P. Sloan, C. F. Rose, and M. F. Cohen. “Shape by example.” *Proc. ACM SIGGRAPH Symp. on Interactive 3D Graphics,* pp. 135-143, 2001.

[79] D. Terzopoulus and K. Waters. “Physically-based facial modeling, analysis and animation.” *Journal of Visualization and Computer Animation* , 1(2):73-80, 1990.

[80] USF DARPA HumanID 3D Face Database, Courtesy of Prof. Sudeep Sarkar, University of South Florida, Tampa, FL.

[81] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. “Face transfer with multilinear models.” *Proc. SIGGRAPH’05,* pp. 426-433, August 2005.

[82] Y. Wang, X. Huang, C. S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang. “High resolution acquisition, learning and transfer of dynamic 3-D facial expressions.”*Proc. EUROGRAPHICS’04,* pp. 677-686, 2004.

[83] K. Waters. “A muscle model for animating three-dimensional facial expression.” *Proc. SIGGRAPH’87,* pp. 17-24, July 1987.

[84] L. Williams. “Performance-driven facial animation.” *Proc. SIGGRAPH’90,* pp. 235-242, August 1990.

[85] J. K. Wu, Y. H. Ang, P. C. Lam, S. K. Moorthy, and A. D. Narasimhalu. “Facial image retrieval, identification, and inference system.” *Proc. 1st ACM International Conference on Multimedia,* pp. 47-55, 1993.

[86] Y. Wu, P. Kalra, L. Moccozet, and N. M. Thalmann. “Simulating wrinkles and skin aging”, *The Visual Computer,* 15(4): 183-198, 1999.

[87] A. Young and D. Hay. *Configurational information in face perception.* Experimental Psychology Society, January 1986.

[88] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. “Space-time faces: high resolution capture for modeling and animation.” *Proc. SIGGRAPH’04,* pp. 548-558, August 2004.

[89] Q. Zhang, Z. Liu, B. Guo, D. Terzopoulos, H.-Y. Shum. “Geometry-driven photorealistic facial expression synthesis.” *IEEE Transactions on Visualization and Computer Graphics,* 12(1):48-60, January 2006.

[90] Y. Zhang. “An efficient texture generation technique for human head cloning and morphing.” *Proc. International Conference on Computer Graphics Theory and Applications,* pp. 85-92, 2006.

[91] Y. Zhang and T. Sim. “Realistic and efficient wrinkle simulation using an anatomy-based face model with adaptive refinement.” *Proc. Computer Graphics International’05,* pp. 3-10, 2005.

[92] Y. Zhang, E. C. Prakash, and E. Sung. “A new physical model with multi-layer architecture for facial expression animation using dynamic adaptive mesh.” *IEEE Transactions on Visualization and Computer Graphics*, 10(3):339-352, May 2004.

A Anthropometric Landmarks Used for Our Method

Table 6 lists the landmarks that are used for the region-based shape synthesis method described in this paper. We follow naming conventions used in the craniofacial anthropometry literature [31, 32]. Reference should be made to Fig. 10 that illustrates the landmarks.

ID	Name	Definition
Eyes		
sci	Superciliare	The highest point on the upper margin of the middle portion of the eyebrow
fz	Frontozygomaticus	The most lateral point on the frontozygomatic suture
ft	Frontotemporal	The most medial point on the temporal crest of the frontal bone
en	Endocanthion	The inner corner of the eye fissure where the eyelids meet
ex	Exocanthion	The outer corner of the eye fissure where the eyelids meet
pi	Palpebrale inferius	The lowest point in the middle of the margin of the lower eyelid
ps	Palpebrale superius	The highest point on the upper margin of the middle portion of the eyelid
Nose		
n	Nasion	The midpoint of the nasofrontal suture
se	Sellion	The deepest point of the nasofrontal angle
mf	Maxillofrontal	The anterior lacrimal crest of the maxilla at the frontomaxillary suture
prn	Pronasal	The most protruded point of the nasal tip
sn	Subnasal	The junction between the lower border of the nasal septum
sbal	Subnasal	The point on the lower margin of the base of the nasal ala where the ala disappears into the upper lip skin
al	Alare	The most lateral point on the nasal ala
Mouth		
ls	Labial superius	The mid point of the vermilion border of the upper lip
li	Labial inferius	The mid point of the vermilion border of the lower lip
sto	Stomion point buccal	The mid point of the labial fissure when the lips are closed naturally
ch	Cheilion	The outer corner of the mouth where the outer edges of the upper and lower vermilions meet
cph	Crista philtre	The point on the crest of the philtrum, the vertical groove in the median portion of the upper lip
ls’	Labiale superius lateralis	The point on the upper vermilion border directly inferior to Subalare (sbal)
Chin		
sl	Sublabial	The midpoint of the Labiomental sulcus
pg	Pogonion	The most anterior point in the middle of the soft tissue chin
gn	Gnathion	The lowest point in the midline on the lower border of the chin
go	Gonion	The most lateral point at the angle of the mandible
go’	Gonion lateralis	The middle lateral point at the angle of the mandible

Table 6: Anthropometric landmarks employed for region-based face shape synthesis.

B Subregion Shape Blending

The position of a vertex $\mathbf{x}_i$ in the feature region F after deformation is $\mathbf{x}'_i$. Let $\mathcal{V}$ denote the set of vertices of the head mesh. For smooth blending, positions of the subset $\overline{\mathcal{V}_F} = \mathcal{V} \setminus \mathcal{V}_F$ of vertices of $\mathcal{V}$ that are not inside the feature region should be updated with deformation of facial features. For each vertex $\mathbf{x}_j \in \overline{\mathcal{V}_F}$, the vertex in each feature region that exerts influence on it, $\mathbf{x}^F_{k_i}$, is the one of minimal distance to it. It is desirable to use geodesic distance on the surface, rather than Euclidean distance to measure the relative positions of two mesh vertices. Given that the head model is represented by an irregular, triangular mesh, we adopt an approximation of the geodesic distance based on a cylindrical projection which is preferable for regions corresponding to a volumetric surface (e.g., the head). The cylindrical projection defines a mapping between the 3D coordinates of the head mesh and a 2D image plane. The distance between two vertices of the projected mesh in the 2D image plane is a fair approximation of geodesic distance. Thus, $\mathbf{x}^F_{k_i}$ is obtained as:

$$\| \mathbf{x}_j - \mathbf{x}^F_{k_i} \|_G = \min_{\{i|i\in\mathcal{V}_F\}} \| \mathbf{x}^*_j - \mathbf{x}^*_i \| \tag{10}$$

where $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ are the positions of vertices on the projected mesh, and $\| \cdot \|_G$ denotes the geodesic distance. Note that the distance is measured offline in the original undeformed generic mesh. For each non-feature vertex $\mathbf{x}_j$, the displacement vector for its corresponding closest feature vertex $\mathbf{x}_{k_i}^F$ is used to update its position in shape blending:

$$\mathbf{x}'_j = \mathbf{x}_j + \sum_{F \in \Gamma} \exp(-\frac{1}{\alpha} \| \mathbf{x}_j - \mathbf{x}_{k_i}^F \|_G) \| {\mathbf{x}'_{k_i}}^F - \mathbf{x}_{k_i}^F \| \qquad (11)$$

where Γ is the set of facial features and α controls the size of the region influenced by the blending. We set α to 1/10 of the diagonal length of the bounding box of the head model.

C Subregion Texture Blending

We use region masks to facilitate the blending (see Fig. 15). The pixels on the outermost ring of a feature region are grouped into the *boundary pixel set* $P^0 = \{p_1^0, \ldots, p_{n_0}^0\}$, where n_0 is the number of boundary pixels. We then identify N rings of pixels around the feature region as the blending region. Let $C(P^0) = \{C(p_1^0), \ldots, C(p_{n_0}^0)\}$ and $C'(P^0) = \{C'(p_1^0), \ldots, C'(p_{n_0}^0)\}$ denote the color sets of boundary pixels before and after texture morphing, respectively. The change of boundary pixel colors is used to update colors of the set $P^j = \{p_k^j \mid k = 1, \ldots, n_j\}$ of pixels that are in the jth ring around the region, where j ranges from 1 to N and n_j is the number of pixels in the jth ring. The color updating is executed in a wave-propagation order starting at the 1st ring and expanding towards the Nth ring:

$$C'(p_k^j) = C(p_k^j) + \frac{W^j}{s_k^j} \sum_{l=1}^{s_k^j} (C'(p_l^{j-1}) - C(p_l^{j-1})) \quad j = 1, \ldots, N \qquad (12)$$

where p_l^{j-1} are the pixels that are adjacent to the pixel p_k^j and in the (j-1)th ring, s_k^j is the number of all found pixels p_l^{j-1} for p_k^j, and W^j is a weight function to attenuate the color updating. W^j is defined according to the distance between the pixel and the region boundary measured in terms of the number of the ring in which the pixel is located:

$$W^j = \frac{f(j)}{f(j-1)} \quad j = 1, \ldots, N \qquad \text{where} \quad f(x) = \frac{e^{-\kappa x} - e^{-\kappa N}}{1 - e^{-\kappa N}} \qquad (13)$$

where parameter κ controls the profile of function curve. From a maximal amplitude of 1 corresponding to the pixels in the feature region boundary, f decreases nonlinearly to 0 at the Nth-ring of pixels around the feature region. We use $\kappa = 0.2$ and $N = 10$ in our experiments.

For a fast implementation, since W^j and s_k^j are static for texture blending, we calculate them for pixels in the blending region offline by making use of the adjacency relations that are defined on the cylindrical texture image, such as circulating through the pixels adjacent to a given pixel and identifying the ring a given pixel is located in. The calculated ratio values W^j / s_k^j are coded into the alpha channel of the mask image shown in Fig. 15. In the online simulation, we only calculate color change of the pixels for a quick texture blending according to Eq. 12. As shown in Fig. 15, the blending regions of every two features overlap. To obtain a smooth color transition, we compute a weighted average of the change of the pixel color contributed from each blending region for pixels in the overlapping zone.