Viseme-aware realistic 3D face modeling from range images

Ken Yano and Koichi Harada,

Graduate School of Engineering
Hiroshima University, Hiroshima, Japan

Summary

In this paper, we propose an example based realistic face modeling method with viseme control. The viseme describes the particular facial and oral positions and movements that occur alongside the voicing of phonemes. In facial animation such as speech animation and talking head, etc, a face model with open mouth is often used and the model is animated along with the speech sound by synchronizing the change of mouth shape to that of the phoneme.

We claim that visemes as well as facial expressions are idiosyncratic by nature, so they have to be modeled specific to the subject to generate a truthful facial animation for the subject. The proposed method tries to automate the creation of a realistic face model with viseme control from a set of scanned data.

Key words:
Face modeling, surface reconstruction, morphing

1. Introduction

Creating face models that look and move realistically is an important problem in computer graphics. It is also one of the most difficult ones, as even the minute changes in facial expression can reveal complex moods and emotions. The presence of very convincing characters in recent films makes a strong case that these difficulties can be overcome with the aid of highly skilled animators. Because of the sheer amount of work required to create such models, there is a clear need for more automated techniques.
A viseme is a supposed basic unit of speech in the visual domain. It describes the particular facial and oral positions and movements that occur alongside the voicing of phonemes.

We propose an example based realistic face modeling technique especially to create a morphable face model with expressive viseme control.
The input to our system is a set of range and color images of the specific subject scanned by the scanner [1]. Note that at each pixel coordinate of the range image contains a measured 3D coordinate and each pixel of the color image contains RGB color intensity.
A pair of the two images is scanned simultaneously and has natural “pixel-to-pixel" correspondence between them.

A static 3D face model is generated in two folds. First a general 2D facial template mesh is fitted to the images of the subject using RBF (Radial Basis Function) method. The fitting is processed in 2D space, the image plane, using the constraints set by the user. The constraint is a set of fiducial marker points, such as the tip of the nose and corners of the eyes, etc.
After the RBF fitting, a 3D coordinate is sampled at each vertex of the deformed mesh from the range image as well as a texture coordinate from the color image. The connectivity of the sampled points is naturally given by the template mesh.

The face models generated offer "point-to-point" correspondence among them irrespective of the kind of the viseme of the subject. This feature makes it possible to statistically analyze the geometric changes by viseme.

In facial modeling, sculpting expressive open mouths requires daunting task even for a skilled modeler as well as expressive eyes. Moreover, articulating an open mouth in a realistic way is one of the difficulties that an animator encounters for facial animation.
In order to estimate the geometric changes of a face model due to viseme, especially the shape of a open mouth, we first prepare the 3D face models for a set of visemes as well as the base model (closed mouth position).

Given a set of 3D face models for each viseme and for the closed mouth, the geometric changes due to each viseme are estimated by subtracting the base model from the face model of the viseme. In order to acquire reliable motion vectors, the geometric changes, we create a stencil mask to remove undesired motion vectors for the upper part of the face.

Given the geometric changes for each viseme, we construct a morphable face model with expressive viseme control. Two types of morphing are experimented. The first is the blend shapes in which a novel face model is create as a convex linear combination of n basis vectors, each vector being one of the blendshapes, e.g., the face itself.

In the second method, we apply PCA (Principal Component Analysis) to the face space. PCA is a statistical tool that transforms the space such that the covariance matrix is diagonal (i.e., it decorrelates the data).

Manuscript received April 5, 2009
Manuscript revised April 20, 2009

The morphable model by PCA provides an another alternative to generate a novel face model with expressive viseme, however in most cases the basis vectors of the decomposed face space do not have clear semantics, so it requires the user some times to figure out what effects each basis vector might have for the deformed face model.
The main contribution of this paper is to create a realistic morphable face model with viseme control from a set of scanned data. The proposed method does not require special modeling or animation skills and proceeds with as little as user interventions.
Most of the relevant works pay little attention or no attention about the change of facial geometry due to the open mouth positions or the visemes.
The results ensure that our method provides a novel way to generate a realistic face model with expressive viseme control and the generated models can be used for further processing for downstream applications.

2. Related Works

Correspondent face models are often used in graphics and vision community. Blanz et al [10] create a morphable face model by taking range images of several faces using a 3D scanner and putting them into "one-to-one" correspondence by expressing each shape using the same mesh structure. Using the morphable model, it is possible to group changes in vertex position together for representing common changes in shape among several surfaces. Using principal component analysis, they succeeded to find a basis for expressing shape changes between faces.
Model with full correspondence was also studied by Praun et al [9]. In order to establish full correspondences between models, they create a base domain that is shared between models, and then apply a consistent parameterization. They search for topologically equivalent patch boundaries to create base domain mesh.
Wang et al [8] propose a novel approach for facial expression analysis from images. They decompose facial expression by Higher-Order Singular Value Decomposition (HOSVD), a natural generalization of matrix SVD and learn the expression subspace.
Several papers propose the method to learn facial expression from video [7] [11]. Those works typically involves visual tracking of: facial movement such as contours, optical flow, or transient features such as furrows and wrinkles.
Facial expressions are learned and recognized in relation to Facial Action Coding System (FACS) [6], a popular representation that codes visually distinguishable facial movement in small units. These action units describe qualitative measures such as "pulling lip corners", or "furrow deepening". A facial expression is described as a combination of these units.
Vlasic et al [5] proposes a method for expressing changes in face shape using a multi-linear model, accounting for shape changes not only based on the subject's identity but also based on various expressions and visemes.
Yong et al [12] presents a novel approach to produce facial expression animation for a new model. Instead of creating new facial animation from scratch for each new model created, they take advantage of existing animation data in the form of vertex motion vector. They call this process "expression cloning" and it provides an alternative way for creating facial animation for character models.

3. Realistic 3D face modeling

The techniques of realistic face modeling and animation have been spurred by emergence of 3D scanning devices at relatively low cost.
Since range data contains artifacts such as spikes and holes, it needs to be repaired for further processing. Furthermore only facial part of whole scanned data is relevant, unnecessary portion of the data has to be removed with great care and skills.
The proposed modeling method tries to automate such lengthy operations and produce a realistic face model for specific subject. The details of our method are described as follows.

3.1 Method - 2D Template fitting

The input to our system is range and color images of the subject scanned by range scanner [1]. The two images are taken simultaneously by one scan so there is a natural correspondence between each pair of the pixels of the two images.
At each pixel location of the range data contains a measured 3D coordinate (x,y,z) from the view of the scanner. To display range images, the depth information (z coordinate) of each pixel is converted to gray scale intensity.

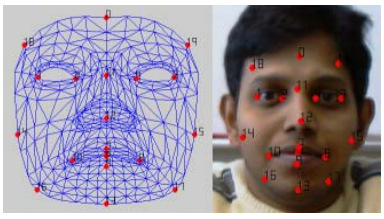


Figure 1: From left to right; (a) A 2D template mesh is shown with facial feature points. The template mesh contains 441 vertices and 804 faces. (b) The user is asked to select feature points interactively on the color image. The selected feature points are shown in red. Each selected feature point corresponds to the feature point defined on the 2D template mesh.

First, a 2D template mesh (with zero depth) shown in Figure 1 is prepared on which a set of fiducial feature points (FP) is defined. Twenty feature points in total are empirically defined.
The user is asked to select facial feature points (FP') interactively on color image (Figure 1). As a result, there is a one-to-one mapping, Φ , from the points set (FP) to the point set (FP').

After establishing the mapping, Φ , the template mesh is deformed and fitted to the face image using the mapping as the constraints.
The fitting is done using RBF (Radial Basis Function) technique, a well known scattered data interpolation technique, or some authors call it TPS (Thin Plate Spline) interpolation. The detail of the algorithm as follows.

Given a pair of point patterns with known correspondences $U = (u_1, u_2, ..., u_m)^T$ and $V = (v_1, v_2, ..., v_m)^T$, where $U \subset (FP), V \subset (FP')$, and $m = 20$ in our case, we need to establish correspondences between other points on the template mesh to the pixel location on the facial image.

The warping function, F , which warps the point set U to the point set V subject to perfect alignment, is given by the conditions

$$F(u_j) = v_j$$

for $j = 1,2,...,m$. The interpolation deformation model is given in terms of the warping function $F(u)$, with

$$F(u) = A(u) + \sum_{i=1}^{m} \lambda_i \phi(|u - u_i|), u \in R^2$$

,where $\phi(u) = |r|^2 \log(r)$ is a radically symmetric basis function, which is known as thin plane spline function, λ_i is a real-valued weight, and $|u - u_i|$ is Euclidean distance. The function, $A(u)$, is a degree one polynomial that accounts for the linear and constant portion of $F(u)$ which is defined as $A(u) = a0 + a1 * u_x + a2 * u_y$ for 2D interpolation.
Solving for the weights λ_i and the coefficients of $A(u)$ subject to the given constraints yields a function that both interpolates the constraints and minimizes the Equation above in least-square sense.
In order to solve for the unknown weights λ_i and the coefficients of $A(u)$, the constraints become

$$v_j = A(u_j) + \sum_{i=1}^{m} \lambda_i \phi(|u_j - u_i|)$$

Since the equation is linear with respect to the unknowns, λ_i and the coefficients of $A(u)$, it can be formulated as a linear system.

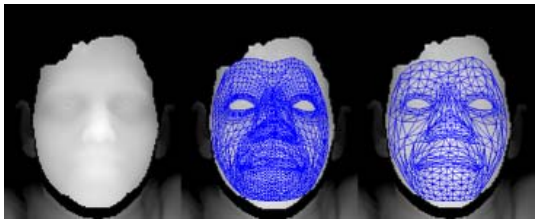


Figure 2: Fitting the 2D template mesh to the image. From left to right; (a) range image (b) deformed 2D template mesh (low resolution) (c) deformed 2D template mesh (high resolution)

The resulting function $F(u)$ exactly interpolates the given facial features (FP') and is not subject to approximation or discretization errors. Note that the number of weights to be determined does not grow with the number of vertices in the template mesh; rather it is only dependent on the number of constraints.

The 2D template mesh is deformed by the function, $F(u)$, and it is shown in Figure 2. As the figure indicates, the template mesh is nicely deformed and fitted to the facial area on the image with all the feature points exactly interpolated.

In order to generate a 3D face model, 3D coordinate at each vertex position of the deformed 2D template mesh is calculated from the range image. As mentioned before, at each pixel coordinate of the range image contains measured 3D coordinate of the subject, then the 3D coordinate is calculated as a bilinear interpolation of the 3D coordinates at the four neighboring pixel coordinate of the vertex position. Note that each vertex of the deformed mesh should not necessarily reside on the grid pixel coordinate of the image; we use bilinear interpolation to correctly calculate the 3D coordinates.

One of the well known issues pertaining to 3D scanned data is that there are holes or missing data. For various

reasons reflected laser beam may be obscured or dispersed so that the sensors see no range or color data at some surface points. Especially for frontal face, those missing data commonly occur on the underside of the jaw, the head hair, the eyebrows, open mouth, and pupils of the eyes, etc.

There are some techniques to fill in those missing data. For example [3] introduces the following algorithm

> [Step 0] The range data is converted to a floating point array with the missing values set to 0.0
> [Step 1] A second matte array is created and set to 1.0 where valid range data exist and to 0.0 where gaps exist.
> [Step 2] In the regions around 0.0 matte value, a small blurring filter kernel is applied to both the matte data and the surface data
> [Step 3] Where the blurred matte value increases above a threshold value, the surface sample is replaced by the blurred surface value divided by blurred matte value.
> [Step 4] The filtering in recursively applied until no matte values are below the threshold

We have integrated the above algorithm to our system and applied to the range image before sampling the 3D coordinates. Note that this algorithm only fills in the small gaps and cannot cope with large missing area such as the head hair and the background.

The resolution of the initial 2D template mesh is considered to be low, so as the resolution of the generated 3D face model. In order to increase the resolution, we apply the one-to-four subdivision method to the deformed 2D template mesh before sampling 3D coordinates. The subdivision method subdivides each triangle face into four triangle faces, thus increases the resolution by the factor of four.

Figure 2 shows the deformed template mesh before and after the subdivision.
Since the color image and the range image have "one-to-one" pixel correspondence between them, as we re-sample 3D coordinates from the range image, we can also obtain texture coordinate at each vertex of the 3D face model.
Figure 3 shows the generated 3D face model in various rendering modes for a subject.

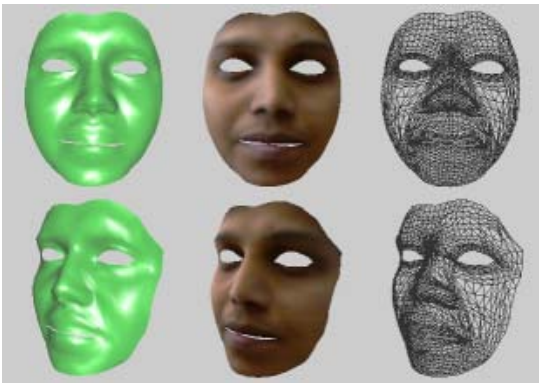


Figure 3: Generated 3D face model. From left to right; (a) shading model (b) texture model (c) wire frame model

4. Modeling of viseme

Modeling a realistic face with open mouth is a challenging task.
In order to create a fully functional face model, it is common that face, teeth and tongue, etc are modeled separately and are placed later at appropriate positions of the entire face model.
Since we only concern about the dynamics of facial geometry due to viseme, we do not discuss the issues of teeth, tongue, etc.

The dynamics of mouth shape is idiosyncratic as well as the facial expression, so we claim that visemes should differ from person to person.
Our modeling method of visemes is an example based and the changes of the geometry of the mouth are learned and modeled from a set of scanned data taken from the same subject from whom we model the base (closed mouth) face model in Section 3.

The details of the method are described as follows.

4.1 Method

Obtain a scanned data for each viseme from the subject. The subject is asked to make natural mouth shape in five positions one by one by pronouncing these five words then halting the mouth position at the underlined two characters (i. car ii. man iii. she iv. eel v. too)

Step1: Generate a 3D face model for each viseme using the same technique in Section 3. However we use a different set of feature points to handle the deformation of the open mouth. The new set of the feature points and the result of the fitting is shown in Figure 4.

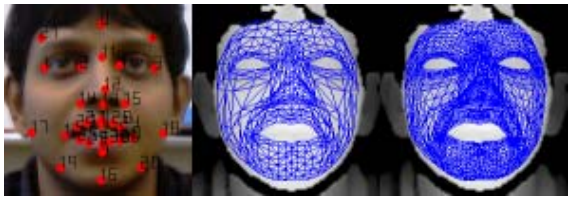


Figure 4: Fitting the 2D template mesh to the image of open mouth. From left to right; (a) color image with markers. A new set of markers are defined to model open mouth (b) deformed 2D template mesh (low resolution) (c) deformed 2D template mesh (high resolution)

Step2: Estimate 3D motion vectors from the base model (closed mouth) to the specific model of the viseme. Since the two models are generated using the same template mesh, there is a natural correspondence between each pair of the vertices.
In order to obtain the 3D motion vectors, we first align the two model (base model and the viseme specific model) using ICP (Iterative Closest Point) method [4].

Step3: The initial correspondences of the points of the two models are given by the user. As shown in Figure 5, six pairs of points are selected for each model. These points are selected because they are not likely to be transformed due to the deformation of the mouth shapes and six points are enough to calculate the rotation and the translation of the rigid transformation between the two models.

Figure 6 shows the alignment of the two models before and after ICP method is applied.

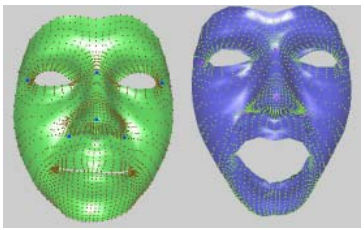


Figure 5: Point correspondence between the two model for ICP

The 3D motion vectors can be obtained from the aligned models by subtracting the base model from the specific model of each viseme. However, we do not want undesired effects of the motion vectors over the upper face due to the changes of the mouth shape; we suppress them by making a stencil mask over the face.

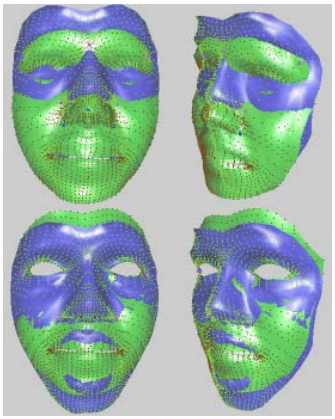


Figure 6: Rigid alignment of the two model by ICP. First row: initial pose of the two models, second row: aligned poses after ICP is applied

The stencil mask is created in these steps.

1. Generate seeds points for the stencil mask. The seeds points are generated by tracing the shortest path between selected two points. The seeds points are all the vertices along the shortest path generated. In order to make a valid mask, a set of five shortest paths is generated using Dijkstra algorithm. The start and the goal point of each shortest path are defined empirically. These steps are illustrated in Figure 7.

2. After obtaining the seed points, we make virtual spheres centered at each seed point then assign every vertices which is inside of one of the spheres to the stencil mask. The radius of the sphere is decided as 20mm for the subject. Figure 7 (d) shows the generated final stencil mask (vertices in orange color).

3. The 3D motion vectors under the stencil mask are suppressed to zero vectors. Compared with the base model, the viseme specific model can reveal noisy measured data in the open mouth area. This is because at the area around the open mouth, the reflected laser beam may be obscured in the hollow area between the lips or dispersed from the teeth. Figure 8 shows unprocessed mesh models from range data and the mesh models generated by our method.

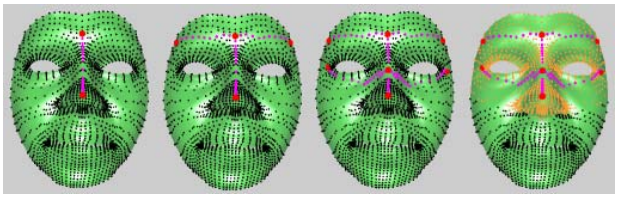


Figure 7: Mask generation to extract 3D motion vector. From left to right, (a) seed points from the first path (b) seed points from the second and the third paths (c) seed points from the fourth and fifth paths (d) generated mask. The mask is the vertices in orange. The uncovered vertices are colored in black.

In order to smooth the noisy 3D motion vectors, we use Laplacian smoothing, e.g., a signal processing approach proposed by [2]. Laplacian smoothing removes high frequency noise. The formula is given as

$$\partial d(v_i)/\partial t = w\Delta d(v_i)$$

$$\Delta d(v_i) = \frac{1}{N(v_i)} \sum_{v_j \in N(v_i)} (d(v_j) - d(v_i))$$

,where $d(v_i)$ is a motion vector at the vertex v_i, $N(v_i)$ is the 1-ring neighbors of the vertex v_i and w is a scaling factor $0 \le w \le 1$. We set $w = 0.1$ and the smoothing operation is iterated thirty times to get the final 3D motion vectors.

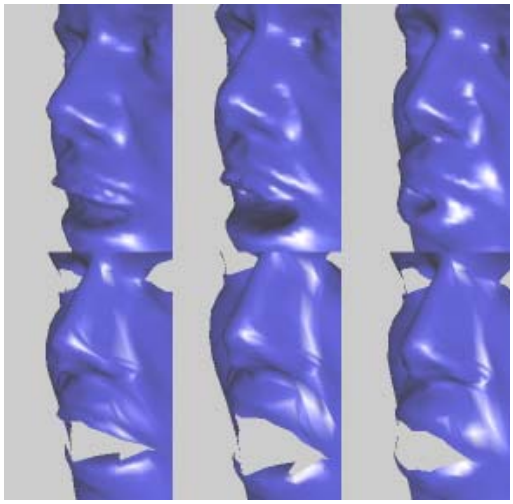


Figure 8: Noisy measured data around the open mouth area. First row shows the unprocessed 3D model obtained from the range data. Second row shows the generated 3D model using the template fitting. Those undesired noisy sampling data around the mouth is the result of noisy scanning data.

4.2 Generated models

Figure 9 shows generated face model of specific viseme. The motion vectors obtained are applied to the base model to get the face model. Figure 10 shows a set of generated face models for each viseme.
The procedure is applied to two different subjects and the results are shown in Figure 17

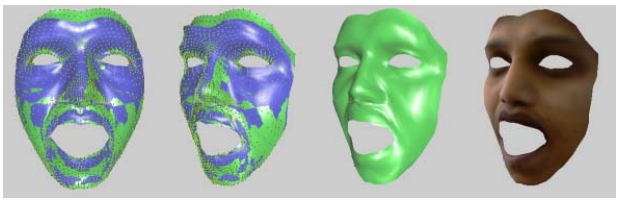


Figure 9: The result of deformation. Target model (blue), deformed model (green)

Figure 10: The result of deformation for all targets. From left to right (a) viseme(car) (b) viseme(she) (c) viseme(man) (d) viseme(eel) (e) viseme(too)

5. Morphable face model

In this section example-based morphable 3D face models are created. Two methods are experimented.
The first method represents the morphable model as a blending of example models. A linear combination of the examples models of specific viseme describes a realistic change of visemes.

The second method applies PCA to the face space. PCA defines a basis transformation from the set of example models to a basis that has two properties relevant to our morphable face model. First, the basis vectors are ordered according to the variances in the dataset along each vector. Second the basis vectors are orthogonal to each other.
The details of two methods are described as follows.

5.1 Morphing by Blend-Shapes

In order to create a morphable face model, a set of the viseme models generated in Section 4 is aligned to the base model using ICP method.
We let S_0 be the shape vector of the base model and S_i be the shape vectors of the model of the viseme i, the morphable face model is represented as

$$S = \sum_{i=1}^{M} w_i S_i + (1.0 - \sum_{i=1}^{M} w_i) S_0 \quad (1)$$

, where w_i are non-negative blending weights and $M(=6)$ is the number of viseme poses.

The shape vectors S_i are formed from the 3D coordinates of surface points:

$$S_i = (x_1, y_i, z_{i,}..., x_n, y_n, z_n)^T$$

Figure 11 shows the gradual deformation from the base model to the models of three specific visemes. Each deformed model is created by changing the weight relevant to the specific viseme and other weights are set to zero.

Figure 11: Gradual deformation of each viseme. The base model is gradually deformed to the target viseme model by changing the weight of the target (w=0.3: left column, w=0.6: center column, and w=0.9: right column). Target viseme model; first row (viseme(car)), second row (viseme(eel)), third row (viseme(too))

The effect of blending of multiple viseme examples is shown in Figure 12.

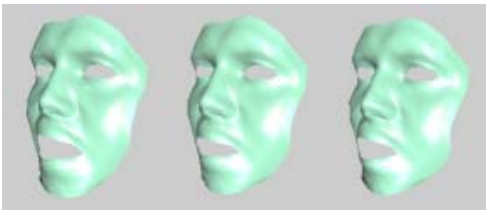


Figure 12: Random blending of various visemes. Left: 0.6 * viseme(man) + 0.3 * viseme(eel) + 0.1 * base, Center: 0.5 * viseme(too) + 0.2 * viseme(car) + 0.3 * base, Right: 0.8 * viseme(too) + 0.1 * viseme(eel) + 0.1 * viseme(car)

5.2 Morphing by PCA

Let a set of aligned 3D face model S_i of each viseme be represented as a matrix X , where

$$X = [S_0 S_1 S_2 .. S_M]$$

,and X is of dimension $3n \times M$, where n is the number of vertices. The difference from the average face model (the sample mean) $\overline{S}$ is the matrix M':

$$X' = [(S_0 - \overline{S})(S_1 - \overline{S})(S_2 - \overline{S})..(S_M - \overline{S})]$$
$$= [S_0' S_1' S_2' ... S_M']$$

Principal component analysis seeks a set of M orthogonal vectors, e_i , which best describes the distribution of the input data in least-square sense, i.e., the Euclidean projection error is minimized. The typical method of computing the principal components is to find the eigenvectors of the covariance matrix C , where

$$C = \sum_{i=0}^{M} S_i' S_i'^T = X'X'^T$$

is $3n \times 3n$. This will normally be a huge matrix, and a full eigenvector calculation is impractical. Fortunately, there are only M nonzero eigenvalues, and they can be computed efficiently with an $M \times M$ eigenvector calculation. It is easy to show the relationship between the two. The eigenvectors e_i and eigenvalues λ_i of C are such that

$$Ce_i = \lambda_i e_i$$

These are related to the eigenvectors $\hat{e}_i$ and eigenvalues μ_i of the matrix $D = X'X'^T$ in the following way:

$$D\hat{e}_i = \mu_i\hat{e}_i,\ X'^T X'\hat{e}_i = \mu_i\hat{e}_i,\ \ X'X'^T X'\hat{e}_i = \mu_i X'\hat{e}_i$$
$$CX'\hat{e}_i = \mu_i X'\hat{e}_i,\ \ C(X'\hat{e}_i) = \mu_i(X'\hat{e}_i),\ \ Ce_i = \lambda_i e_i$$

, showing that the eigenvectors and eigenvalues of C can be computed as

$$e_i = (X'\hat{e}_i),\ \ \lambda_i = \mu_i$$

In other words, the eigenvectors of the large matrix C are equal to the eigenvectors of the much smaller matrix D, premultiplied by the matrix X'. The nonzero eigenvalues of C are equal to the eigenvalues of D.

Once the eigenvectors of C are found, they are sorted according to their corresponding eigenvalues; a larger eigenvalue means that more of the variance in the data is captured by the eigenvector.

Eigenvectors corresponding to the top K eivenvalues define the face space of the visemes.

The result of PCA is described as follows. Figure 13 shows the average model, $\overline{S}$, and Figure 14 shows top five principal eigen face models.

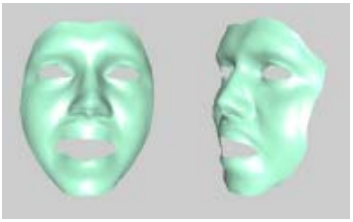


Figure 13: Average face model

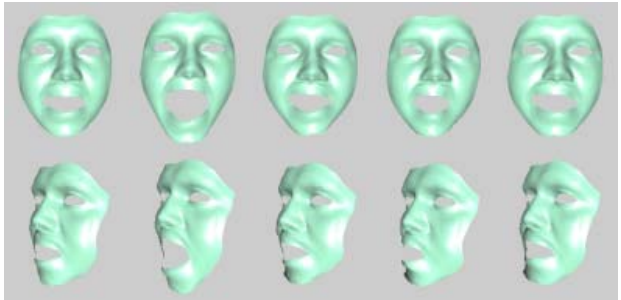


Figure 14: A set of principal eigen face models. From left to right; (a) first eigen model (b) second eigen model (c) third eigen model (d) fourth eigen model (e) fifth eigen model (The eigen models are actually motion vectors, here we add them to the mean face model for display)

The CR (Contribution Rate) and CCR (Cumulative Contribution Rate) of a eigen face model, e_l, are defined as

$$CR = \frac{\lambda_l}{\sum_{j=1}^{M}\lambda_j},\ CCR = \frac{\sum_{i=1}^{l}\lambda_i}{\sum_{j=1}^{M}\lambda_j}$$

Table 1 shows the eigenvalue, CR and CCR of top five principal eigen models. From these results, the CCR of the top three principal models is over 0.95 and the last two principal models are less important. From this analysis, we can conclude that only the top three principal models may be used to define the face space thus compressing the entire model space.

Table 1: The result of PCA analysis

	Eigen value	CR	CCR
First eigen model	124912	0.776	0.776
Second eigen model	20496	0.127	0.903
Third eigen model	8513	0.053	0.956
Fourth eigen model	5596	0.035	0.991
Fifth eigen model	1437	0.0089	1.0

Table 2: PCA coefficients of each viseme model

	ω_1	ω_2	ω_3	ω_4	ω_5
Base	-184.7	-65.2	-33.7	-35.1	-0.1
Viseme(too)	-82.65	105.7	-38.2	16.5	4.4
Viseme(man)	114.0	34.6	22.3	-31.6	-25.0
Viseme(she)	-78.3	8.9	68.9	-3.2	17.8
Viseme(eel)	-22.0	-53.6	5.9	55.5	-13.9
Viseme(car)	253.7	-30.5	-25.2	-2.1	16.8

In order to project each face model to this eigen face space, we first subtract the average face model from the face model then project it into the eigen face space as following

$$(w_1 w_2 ... w_k)^T = (e_1 e_2 ... e_k)^T (S_i - \overline{S})$$

, where w_j is the PCA coefficient of the face model for the ith principal face model e_i.

Table 2 shows the PCA coefficients of each face model and the results are plotted in Figure 15 graphically. From this result, we can assume that the first principal face model describes natural open mouth by jaw rotation, the second principal model describes puckered mouth and the third principal model describes mouth pulled out sideways.

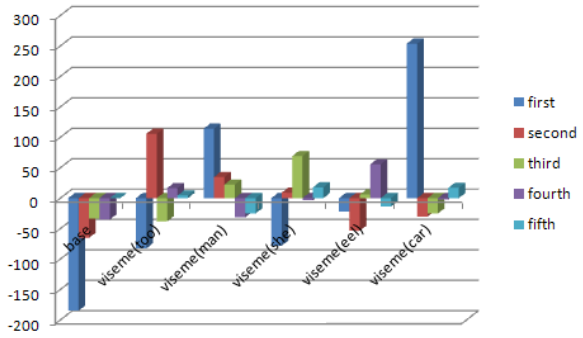


Figure 15: PCA Coefficients of each viseme model. Graphical display of Table 3

Thanks to the face space decomposed by PCA, it gives the user an alternative method to animate the face model or to blend the principal face models. For instance, exaggerated open puckered mouth can be intuitively generated by using the first and second principal face model, which is otherwise difficult to be made by using the method described in Subsection 5.1.

Compared with the Equation 1, the morphable face model by PCA is formulated as

$$S = \sum_{i=1}^{K} \alpha_i \sqrt{\lambda_i} e_i + \bar{S} (-3 \le \alpha_i \le 3) \qquad (2)$$

$\sqrt{\lambda_i}$ is the standard deviation for the ith principal face model. Figure 16 shows example face models generated by blending the top three principal face models. The parameters of the blending are shown in Table 3.

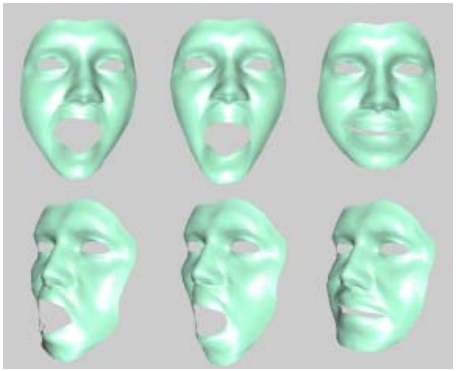


Figure 16: Blending face models generated from the morphable face model by PCA. The parameters of the blending are described in Table 3. From left to right; blend model1, blend model2 and blend model3

Table 3: Morphable model by PCA. The values are the blending parameters, α_i , in Equation 2.

	α_1	α_2	α_3
Blend model 1	0.5	1.0	0.0
Blend model 2	1.0	0.0	-2.0
Blend model 3	-0.5	-0.9	0.9

The modeling by PCA offers better controllability than the former method. Since the basis shapes, the principal face models, are orthogonal with each other, the blending of one basis shape does not interfere with the blending of the other basis shapes. However there is a drawback, e.g., the principal component does not have a clear semantics of the shape itself and require the user some time to get the feeling of the effects that the principal component might have on the result of the deformation.

6. Conclusion

In this paper, we propose an example based face modeling technique especially to create a realistic face model with viseme control. Modeling and animating a realistic human face requires skills and considerable time.

The proposed method lessens the costs required by utilizing the scanned data taken from the subject. The process runs almost automatic except that the user needs to select a set of fiducial markers on the facial images provided by the scanner.

In preparation, we generate a set of face models for the base (closed mouth) and for each viseme.

The face models generated have "point-to-point" correspondence among them and make it straight forward to extract the motion vectors, the displacement vectors from the base model to the specific model of the viseme.

In order to generate a morphable model, two methods are experimented. The first method uses the correspondent face model as the basis of the face space. A novel face model is created by a convex linear combination of the basis vectors.

The other method applies PCA to the face space and a novel face model is created by using the average face model and a set of principal face models. The statistical modeling method by PCA provides another alternative to generate a novel face which cannot be made by the former method.

The proposed method can also be used to extract expression from a subject and generate a morphable face model with realistic expressions. We plan to extend the morphable face model by incorporating the facial expressions to the face model with viseme control, thus provides much greater varieties of facial deformation.

Figure 17: Realistic face models with expressive viseme control for two subjects. Subject1: first two rows, Subject2: second two rows. From the left most column to the right most column { base model, viseme(car), viseme(eel), viseme(man), viseme(she), viseme(too) }

References

[1] MINOLTA, KONIKA. *Vivid 700 : Non-contact 3D laser scanner*. http://se.konicaminolta.us/.

[2] Taubin, Gabriel. *A Signal Processing Approach to Fair Surface Design.* 1995. 22nd annual conference on Computer graphics and interactive techniques.

[3] L.Williams. *Performance driven facial animation.* 1990, Computer Graphics, pp. 235-242.

[4] Mckay, Paul J. Besl and Neil D. *A Method for Registration of 3-D Shapes.* 1992, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 239-256.

[5] Popovic, Daniel Vlasic and Matthew Brand and Hanspeter Pfister and Jovan. *Face Transfer with Multilinear Models.* 2005, ACM Transactions on Graphics, pp. 426 - 433.

[6] Paul Ekman, Wallace V.Friessen. *Facial Action Coding System.*

[7] Blake, B Bascle. *Separability of pose and expression in facial tracking and animation.* 1998. Int. Conf. Computer Vision. pp. 323--328.

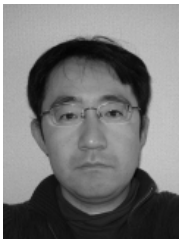
[8] Ahuja,H.Wang. *Facial expression decomposition.* IEEE International Conference on Computer Vision. pp. 958-965.

[9] Schroder, Emil Praun and Wim Sweldens and Peter.*Consistent Mesh Parameterizations.* 2001. Computer graphics and interactive techniques. pp. 179 - 184.

[10] Vetter, Volker Blanz and Thomas. *Face Recognition Based on Fitting a 3D Morphable Model.* 2003, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1063 - 1074.

[11] Bregler, E S Chuang and H Deshpande. *Facial expression space learning.* 2002. Pacific Conference on Computer Graphics and Applications. pp. 68--76.

[12] Ulrich, Jun Yong and Noh. *Expression cloning.* 2001. 28th annual conference on Computer graphics and interactive techniques. pp. 277-288.



Ken Yano is a PhD student of the Graduate School of Engineering at Hiroshima University after working at Sharp Corporation. He received the BE in 1992 from Waseda University and MS in 1998 from Central Michigan University in U.S. His current research is mainly in the area of computer graphics and computer vision. Special interests include 2D and 3D data processing and machine learning. He is a member of ACM, IEEE, IPS of Japan.

Koichi Harada is a professor of the Graduate School of Engineering at Hiroshima University. He received the BE in 1973 from Hiroshima University, and MS and PhD in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research is mainly in the area of computer graphics. Special interests include man-machine interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is a member of ACM, IPS of Japan, and IEICE of Japan.