EUROGRAPHICS 2003 / P. Brunet and D. Fellner
(Guest Editors)

Volume 22 (2003), Number 3

Reanimating Faces in Images and Video

V. Blanz,[1] C. Basso,[2] T. Poggio[3] and T. Vetter[2]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[2] University of Basel, Departement Informatik, Basel, Switzerland
[3] Center for Biological and Computational Learning, MIT, Cambridge (Mass.), USA

Abstract

This paper presents a method for photo-realistic animation that can be applied to any face shown in a single image or a video. The technique does not require example data of the person's mouth movements, and the image to be animated is not restricted in pose or illumination. Video reanimation allows for head rotations and speech in the original sequence, but neither of these motions is required.

In order to animate novel faces, the system transfers mouth movements and expressions across individuals, based on a common representation of different faces and facial expressions in a vector space of 3D shapes and textures. This space is computed from 3D scans of neutral faces, and scans of facial expressions.

The 3D model's versatility with respect to pose and illumination is conveyed to photo-realistic image and video processing by a framework of analysis and synthesis algorithms: The system automatically estimates 3D shape and all relevant rendering parameters, such as pose, from single images. In video, head pose and mouth movements are tracked automatically. Reanimated with new mouth movements, the 3D face is rendered into the original images.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

In terms of photo-realism, the most advanced examples of talking faces so far have been produced with image-based methods [7, 12, 16, 11]. Re-arranging frames in video sequences, Video Rewrite [7] reanimates existing footage to make a person utter new text. To reduce the number of frames to be stored, other methods morph between keyframes [12] of visemes, which are the visual analogue of phonemes. A sophisticated statistical analysis of video footage has yielded other fundamental mouth shapes that can be encoded as a vector space of warp-fields and textures [11]. With iteratively optimized trajectories, this has produced highly realistic speech.

The realism of 2D methods, however, comes at a price: For the person to be animated, images of all basic mouth shapes have to be available, since their appearance is not inferred from other individuals. The output is restricted in pose and other imaging conditions to what is found in the original video: Only small rotations can be covered so far [7, 16], assuming the mouth region to be flat. The gradual occlusions of the teeth by the lips pose additional difficulties to 2D morphing.

In 3D animation, rotations and occlusions are straightforward to achieve. One class of methods involves manually designed deformation patterns of a 3D mesh [25, 26, 29, 2]; Free Form Deformations have been used to animate a person's face, given a front and a side view [15], or multiple stereo-pairs or video frames [13]. An alternative approach is to simulate the physics of surface deformations caused by muscle forces [33, 31, 23, 20]. Given a neutral 3D range scan [31, 23, 20] or CT-scan [22], the physical model can predict that person's facial expressions, and animate the face. In all these techniques, it may be difficult to define deformation patterns, muscles and tissue parameters that produce precisely the wrinkles found on faces. In contrast, example-based methods try to learn deformations from real faces.

A number of example-based 3D methods analyze video data from multiple viewpoints to estimate 3D shape of fundamental expressions [18, 27, 28, 30] or to learn the dynamics of speech [10, 6]. Other methods have used either static 3D scans of closed-mouth expressions [32, 5], or time-sequences of structured-light scans [21]. Unlike performance-driven animation, all these techniques produce novel sequences, rather than reproducing motion in 3D. Some systems can also

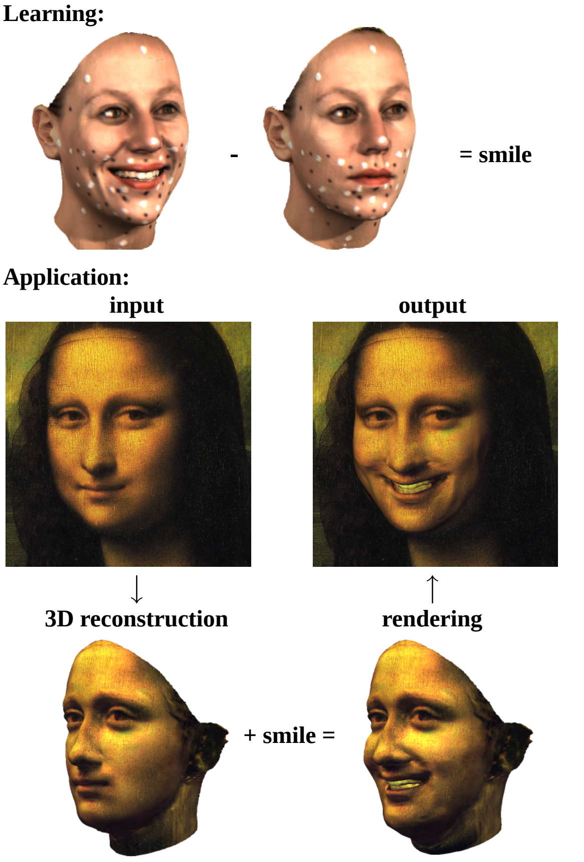


Figure 1: *In the vector space of faces, facial expressions are transferred by computing the difference between two scans of the same person (top row), and adding this to a neutral 3D face. To modify Leonardo's Mona Lisa (second row), we reconstruct her 3D face (third row), add the expression, and render the new surface into the painting (second row, right).*

transfer motion to a novel, neutral face [18, 10, 32, 5, 6, 30], while others transfer high-level parameters, but not the appearance of expressions [27, 28]. Speech and expression can be applied to single images [32, 5, 6, 30] or video [28].

The main contribution of this paper is a framework that combines the strengths of previous animation techniques: the photo-realistic quality of 2D animation, the versatility of a 3D model, the capacity to generate facial expressions of individuals from their neutral faces, and the automated learning technique of example-based methods.

What makes our framework stand out from existing technologies are the low requirements with respect to the input data of the person to be animated: This may be a single image or a video sequence, taken at a wide range of illumination conditions, poses, and mouth shapes. Unlike other methods, we compensate for rotation and speech in video, yet do not need them to animate a given face. This flexibility is crucial for a wide range of applications, such as movie dubbing.

Our method is based on a common vector space of 3D shapes and textures computed from a dataset of 35 laser scans of facial expressions, and neutral faces of 200 persons. In this vector space, expressions can be changed continuously along any trajectory in face space, and transferred across individuals. An estimate of 3D shape from a single image or a video frame is obtained by a fitting algorithm that minimizes the image difference between the synthetic image, and the input image. The algorithm is more general and more robust than previous systems [5], and it can also be applied to non-neutral faces. In that case, we can estimate the neutral shape of the face. For reanimation, we apply new facial expressions to the 3D face, and render it back into the original image or video.

The new vector space representation of open-mouth scans is an extension of previous work on closed-mouth faces and facial expressions [32, 5]. For open mouths, constructing a vector space is significantly more difficult, and has called for additional techniques. Recently, other methods have formed vector spaces of facial expressions from snapshots of dynamic sequences [30, 21, 28]. While some of them[30, 21] are based on 3D coordinates of sparse feature points (64 and 124, respectively), our face vectors from static scans include all vertices of a high–resolution mesh that captures wrinkles and other subtle, yet highly expressive details.

To reanimate faces in video, we are tracking head rotation in the presence of speech and facial expressions. Unlike methods based on facial features[18, 31, 15] or constrained optic flow [8], we minimize image difference in an iterative analysis-by-synthesis loop [5, 30, 14, 28].

In the following section, we introduce the vector representation for faces with open mouths, and a method to construct this vector space. In Section 3, we describe how the model can be applied to animate faces in single images, and show a set of results. Section 4 presents additional methods required for video reanimation.

2. A Morphable Model of Mouth Configurations

The Morphable Model of 3D faces [5] is a vector space of 3D shapes and colors (reflectances). The vectors are defined such that any linear combination of examples

$$\mathbf{S} = \sum_{i=1}^{m} a_i \mathbf{S}_i, \quad \mathbf{T} = \sum_{i=1}^{m} b_i \mathbf{T}_i. \tag{1}$$

is a realistic face, given that $\mathbf{S}$, $\mathbf{T}$ are within a few standard deviations from their averages. In this paper, each vector $\mathbf{S}_i$ is the 3D shape of a human face, stored in terms of x, y, z-coordinates of all vertices $k \in \{1, \ldots, n\}$ of a high-resolution 3D mesh:

$$\mathbf{S}_i = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T. \tag{2}$$

In the same way, we form texture vectors from the r, g, b surface colors of all vertices:

$$\mathbf{T}_i = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \tag{3}$$

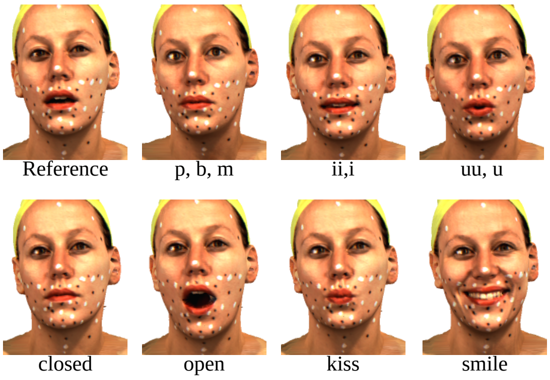


Figure 2: *Examples from the dataset of 35 static 3D laser scans forming the vector space of mouth shapes and facial expressions. 17 scans show different visemes, others show the mouth opening gradually.*

Face space provides a representation not only for different persons' faces [5], but also for changes within one face, as the person speaks or acts. In this paper, we construct a vector space of facial movements and facial expressions recorded from one person, and combine it with the vector dimensions of personality.

For animation, smooth motions are generated by any continuous trajectory in $a_i, b_i \in \mathbb{R}$. This property, however, does not prevent structures, such as eyebrows, from disappearing and reappearing somewhere else on the surface during transitions. To avoid such artefacts, vector components x_k, y_k, z_k have to represent the same structure, such as the corner of an eyebrow, in all vectors $\mathbf{S}_i$. We describe an algorithm to establish this point–to–point correspondence in Section 2.3.

In our representation, any snapshot of a person's face can be mapped to a vector $\mathbf{S}$, $\mathbf{T}$. Even though faces are controlled by a relatively small number of parameters, we cannot expect the full range of expressions to lie within the span of a few extreme shape vectors only, due to the non-linear physical properties that are involved in facial expressions, causing effects such as wrinkles Therefore, we increase the number of dimensions by including intermediate expressions as additional basis vectors.

2.1. Database of Expressions and Mouth Shapes

In order to capture the degrees of freedom of mouth movements for speech synthesis, we recorded a set of 35 static laser scans (Figures 2, 3) of one person. The dataset contains the visemes that will be used as morph-targets in animation, and additional scans that vary systematically in the vertical opening of the mouth, and the width of the mouth. We recorded two additional scans (Figure 3) that display most of the upper and lower jaw teeth. Even though markers are dispensable for our algorithm, we painted white and black spots on the skin to measure tangent motion along the surface (cheeks), and achieve more precise 3D alignment (forehead). Red lipstick increased the contrast at the edge of the lips, and a bathing cap covered the hair.

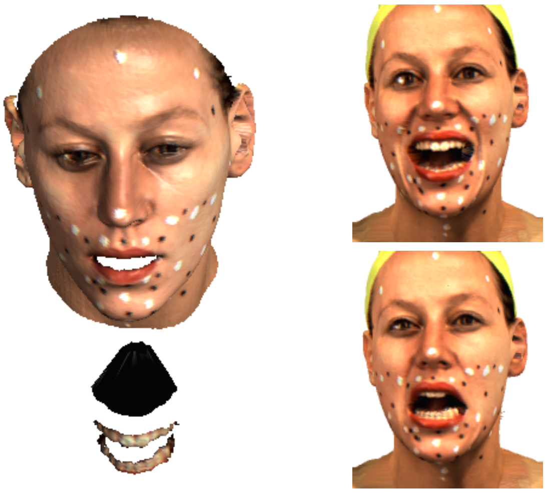


Figure 3: *The reference shape, consisting of the face and lips (left, top), the inner part of the mouth (left, center), and the teeth (bottom). Upper and lower jaw teeth were taken from two different scans (right).*

The 3D scans were recorded with a *Cyberware*^{TM} 3030PS laser scanner. In 512 steps in height h and azimuth ϕ, the scanner records radius $r(h,\phi)$ and coloured texture $R(h,\phi)$, $G(h,\phi)$, $B(h,\phi)$.

2.2. Reference Surface

The morphable model is based on a reference surface mesh. From this surface, point-to-point correspondence to all other scans is established. For the selection of the reference shape, two issues have to be considered: (1) To be able to establish correspondence, the reference surface has to be as similar to the other scans as possible. (2) Only the surface regions that are part of the reference face can be represented in novel linear combinations. The reference mesh has to contain whatever portion of the teeth is visible in speech and facial expressions.

To fulfill the first requirement, we selected an intermediate mouth configuration (Figure 2) as a reference. Much of the teeth is occluded in this shape, so we added teeth to the referenced mesh in a later processing step (Section 2.4). The combined reference mesh (Figure 3) has 90831 vertices at a spacing of about 0.6 mm.

2.3. Correspondence between 3D Scans

The crucial step in forming a morphable model from a set of surface scans is to identify corresponding points on the example scans for all vertices of the reference mesh. To establish dense point-to-point correspondence on the entire sur-

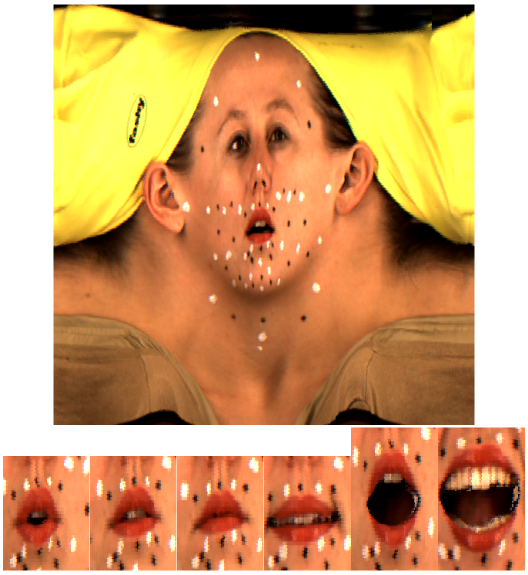


Figure 4: *Top: Texture of the reference scan. Bottom: Textures of mouth configurations; Occlusions of teeth, tongue and pharynx make it difficult to identify corresponding points.*

face of the face, we compute the best match for all structures, rather than using a sparse set of features, or markers. Our algorithm uses both shape and texture. We do not match the teeth, since they are connected to the skull and to the lower jaw, so their motion can be simulated directly (Section 2.4). Still, the fact that teeth, tongue and pharynx are visible in some scans and occluded in others (Figure 4) makes the computation of correspondence a more challenging task.

Unlike the fully automated procedure for neutral faces [5], we have partitioned the scans into 3 batches, depending on how similar they are to the reference. We then perform the following bootstrapping method with minor manual interaction:

The 11 scans of the first batch are reliably processed by an automated algorithm based on optical flow [1]: When applied to grey-level images $I(x,y)$, $I'(x',y')$, optical flow algorithms compute correspondences $(x,y) \mapsto (x',y')$. We use a generalization[5] of this to match laser-scans $r(h,\phi)$, $R(h,\phi)$, $G(h,\phi)$, $B(h,\phi)$ (Section 2.1) with the reference scan; The algorithm finds a mapping $(h,\phi) \mapsto (h',\phi')$ that minimizes the sum of square differences of radii and colors

$$(r(h,\phi) - r'(h',\phi'))^2 + (R(h,\phi) - R'(h',\phi'))^2 + \\ (G(h,\phi) - G'(h',\phi'))^2 + (B(h,\phi) - B'(h',\phi'))^2 \qquad (4)$$

in each point (h,ϕ). Based on this mapping, the scans are converted into shape vectors $\mathbf{S}_i$ (Equation 2).

Principal Component Analysis (PCA[9]) provides an orthogonal basis $\mathbf{u}_i$ adapted to the statistics of the examples $\mathbf{S}_i$, with basis vectors ordered according to the variance in the dataset around the arithmetic mean $\bar{\mathbf{u}}$. For the next step, it is important that linear combinations $\mathbf{S} = \bar{\mathbf{u}} + \sum \gamma_i \cdot \mathbf{u}_i$ can produce shapes beyond the convex hull of examples, with mouths more open or closed than those in batch 1.

For each scan in batch 2, we approximately reproduce mouth shape by adjusting the coefficients γ_i in an interactive tool. Correspondence from these closest linear combinations to the original scans of batch 2 is then computed automatically by the optic-flow-based algorithm, and batch 2 is added to the vector space. Another iteration of this procedure adds batch 3 to the space. In the last bootstrapping iteration, we include teeth to the vector space representation, as described in the following section, to make the computation of optical flow easier (see Figure 4).

2.4. Teeth

In the scans shown in Figure 2, part of the teeth is occluded by the lips. We therefore recorded two scans where most of the teeth are visible (Figure 3), and manually extracted the polygons of the teeth, using an interactive tool. These polygons were then added to the reference surface (Figure 3).

The motion of teeth is easy to simulate: The upper jaw teeth are connected to the upper part of the head, and the lower jaw teeth are connected to the tip of the chin. We exploit these facts in the following way: We align all heads in space using the method of 3D-3D Absolute Orientation [17], based on sets of corresponding points on the upper part of the face, which were located in Section 2.3. Keeping the upper jaw teeth always at the position they have in the scan in Figure 3 (top, right) will then produce correct results for all linear combinations of the example scans.

The lower teeth's motion due to small rotations of the jaw can be approximated by a linear 3D translation: In the original scan (Figure 3, bottom, right), we measure the position of the teeth relative to a point on the tip of the chin. We locate this point in all other scans using correspondence, and shift the teeth to keep their relative position unchanged.

Finally, we add some polygons for the inner part of the mouth extending from the lips back to the pharynx, and intersecting some polygons of the teeth. In each scan, the frontal edges of this surface are connected to the lips.

2.5. Combination of Personality and Expression

Recorded from a single person, the expressions and mouth movements can be transferred to another person's neutral face by simple vector space operations (Figure 1). This procedure assumes that the 3D displacements of surface points are the same for all individuals: We ignore the slight variations across individuals that depend on the size and shape of faces, characteristic patterns of muscle activation, and mechanical properties of skin and tissue. Therefore, our predictions only approximate the true expressions of novel faces.

To be able to transfer facial expressions, we combine the expression vectors with the face vectors of 200 individual neutral faces[5]. The neutral face vectors have to be converted, since they were based on a different, closed-mouth reference surface. Using the correspondence algorithm (Section 2.3), we match this reference scan of personality space to the closed mouth vector in expression space (Figure 2), to find a point-to-point mapping between the two representations. With this mapping, we can automatically resample all individuals' shape and texture data to obtain **S** and **T** in the new format.

Information about shapes and positions of the 200 persons' teeth is unavailable, so we insert the same set of teeth (Section 2.4) behind everyone's closed lips. They are located at a fixed position relative to the center of mass of three points (the corners of the mouth and the center of the lipline) which are located automatically, based on correspondence.

Within the common vector space, Principal Component Analysis could be computed on the combined set of p persons $\mathbf{S}_i^{\mathcal{P}}$ and m expressions $\mathbf{S}_i^{\mathcal{E}}$ simultaneously. However, the relative weight of variances caused by personality and mouth movements, respectively, would depend on the numbers p and m, and this would affect the result of PCA considerably. We therefore keep both sets separate, which yields an average shape $\bar{\mathbf{s}}$ and $p-1$ eigenvectors $\mathbf{s}_i$ for personality, and an average shape $\bar{\mathbf{u}}$ and $m-1$ eigenvectors $\mathbf{u}_i$ for expressions. We use texture eigenvectors $\mathbf{t}_i$ from the personality set only.

We investigated the relation between the subspaces generated by these two sets, by examining their variabilities relative to the averages. Given the sets $\mathcal{E} = \{\mathbf{S}_i^{\mathcal{E}} - \bar{\mathbf{u}} | i = 1 \ldots m\}$ and $\mathcal{P} = \{\mathbf{S}_i^{\mathcal{P}} - \bar{\mathbf{s}} | i = 1 \ldots p\}$, we computed the angle[4] between the subspaces generated by $\mathcal{E}$ and $\mathcal{P}$. The resulting angle of 74° (and of 45° for a control set of identity vectors), shows that the two subspaces are not orthogonal. This characteristic is also reflected by the ratio between the variances (computed as the sum of the eigenvalues of the covariance matrix) of $\mathcal{R}$ and $\mathcal{E}$, with

$$\mathcal{R} = \{(\mathbf{S}_i^{\mathcal{E}} - \bar{\mathbf{u}}) - \sum_{j=1}^{p-1} \frac{(\mathbf{S}_i^{\mathcal{E}} - \bar{\mathbf{u}}) \cdot \mathbf{s}_j}{\|\mathbf{s}_j\|^2} \mathbf{s}_j | i = 1 \ldots m\}$$

being the sets of residuals of the projection of the $(\mathbf{S}_i^{\mathcal{E}} - \bar{\mathbf{u}})$ on $\mathcal{P}$. For the expression set the ratio is 0.46, compared with 0.15 for the control set.

3. Animating Faces in Still Images

In many applications, it is not sufficient to be able to animate a given 3D mesh: First, we may not have a 3D scan of the face, but only one or several 2D images. Second, photorealistic animation often involves re-inserting the moving face into the original scene. By fitting the Morphable Model of 3D faces to the images, we address both aspects of this problem: From a single image of a person, we estimate a textured 3D surface, along with all relevant parameters required to render the modified face back into the original image.

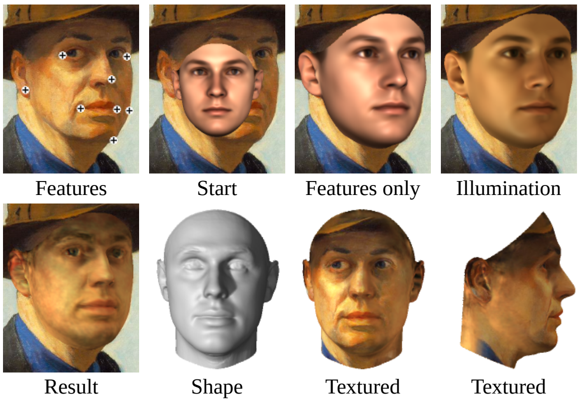


Figure 5: *Recovering a 3D face from E. Hopper's self-portrait: Initialized with manually labeled features (top, left) and starting from a front view of the average face, the algorithm automatically optimizes shape and texture of the morphable model, and estimates pose, illumination, and other parameters. The second row shows the result without (left) and with (right) texture extraction.*

3.1. 3D Reconstruction of Non-Neutral Faces

Based on the combined vector spaces of personality and mouth movements, we estimate 3D shape from images of non-neutral faces, extending an algorithm for neutral faces[5]. In an analysis-by-synthesis loop, the algorithm computes the optimal linear combination of principal components for individual shape $\mathbf{s}_i$, texture $\mathbf{t}_i$, and expression $\mathbf{u}_i$:

$$\mathbf{S} = \bar{\mathbf{s}} + \sum_{i=1}^{p-1} \alpha_i \cdot \mathbf{s}_i + \sum_{i=1}^{m-1} \gamma_i \cdot \mathbf{u}_i, \quad \mathbf{T} = \bar{\mathbf{t}} + \sum_{i=1}^{p-1} \beta_i \cdot \mathbf{t}_i \qquad (5)$$

The estimate is based on an iterative minimization of the difference E_I between the synthetic image $(I_r, I_g, I_b)_{model}$ of the 3D face, and the input image $(I_r, I_g, I_b)_{input}$:

$$E_I = \sum_x \sum_y \sum_{c \in \{r,g,b\}} (I_{c,input}(x,y) - I_{c,model}(x,y))^2. \qquad (6)$$

If multiple images are available, E_I is the sum of all image differences. Minimization is achieved by stochastic gradient descent, evaluating only a random subset of pixels at each iteration.

Along with α_i, β_i, γ_i, the system automatically optimizes all relevant imaging parameters: Three angles for pose, 3D position, focal length of the camera, red, green, and blue intensities of ambient and parallel light for one light source, the direction of parallel light, color contrast, and gains and offsets of the three color channels, which account for the tone and contrast. Unlike previous algorithms [5] that did not optimize focal length and illumination direction, and involved manual pre-alignment of the average face for initial-

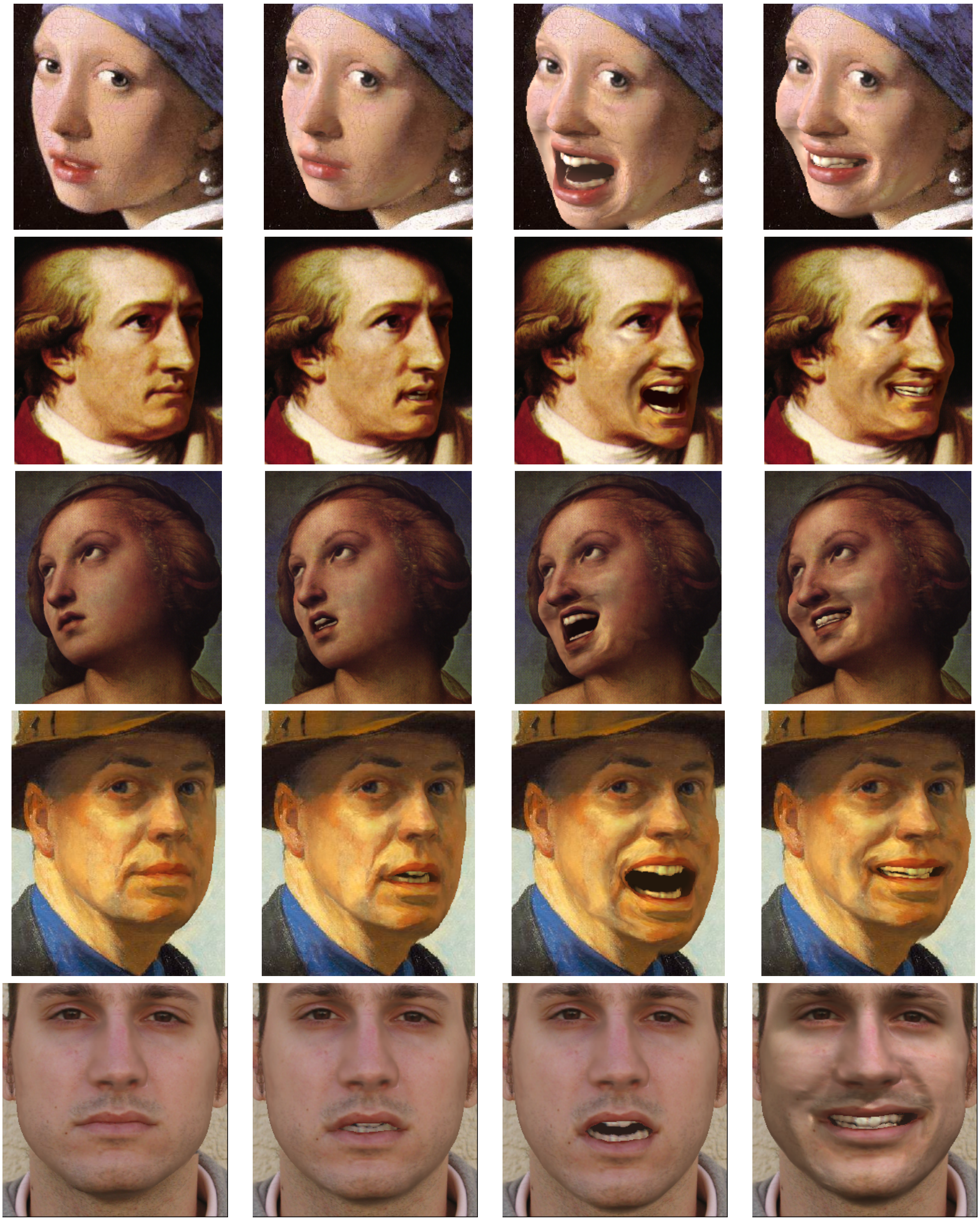
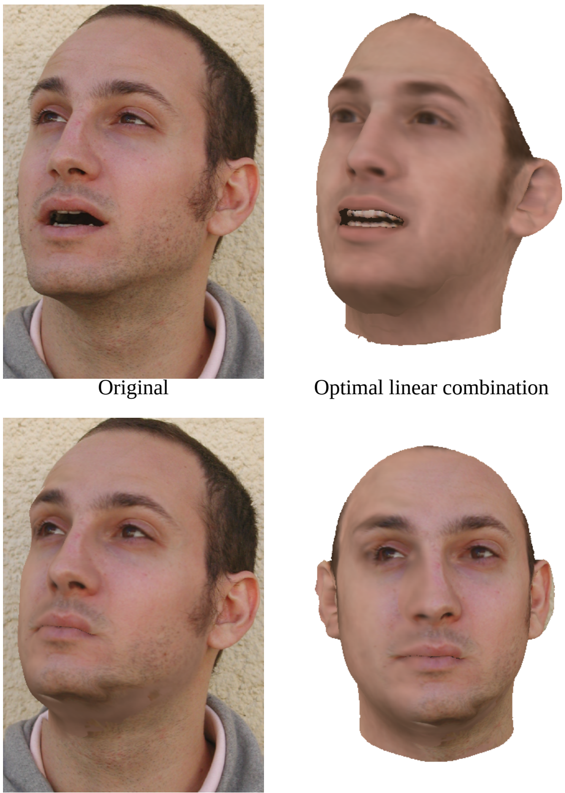


Figure 6: *Reconstructed from the original images (left column), 3D shape can be modified automatically to form different mouth configurations. The paintings are Vermeer's "Girl with a Pearl Earring", Tischbein's Goethe, Raphael's St. Catherine, and Edward Hopper's self-portrait. The bottom left image is a digital photograph. The wrinkles are not caused by texture, but entirely due to illuminated surface deformations. In the bottom-right image, they are emphasized by more directed illumination. Teeth are transferred from 3D scans (Figure 3). Note the open mouth in Vermeer's painting, closed by our algorithm (top row, second image).*



Original

Optimal linear combination

Synthetic neutral

Synthetic neutral

Figure 7: *Top row: 3D reconstruction from an open-mouth image. Among the 11 feature points provided for initialization, only one was in the mouth region (on the upper lip). Therefore, the algorithm must have relied on generic image information to estimate mouth shape. The teeth are not involved in matching currently. Bottom row: Setting the mouth shape coefficients $\gamma_i = 0$ generates a neutralized face (the true neutral face is shown in Figure 6.)*

ization, the new system always starts with a frontal view at standard size, position, and illumination.

For initialization, the user selects between 7 and 20 feature points in the image, such as the corners of the eyes, and clicks on the corresponding vertices of the 3D mesh. Feature points may also be selected along occluding contours, such as the cheek (Figure 5). For these, the algorithm finds temporary correspondences that change during optimization as the face rotates and deforms: It assigns the point in the image to the closest vertex among those with a surface normal that is orthogonal to the line of sight.

The image coordinates $(q_{x,j}, q_{y,j})$ of feature points j contribute to the cost function in the following way: Let (p_{x,k_j}, p_{y,k_j}) be the image positions of the corresponding vertices k_j predicted by the model at the current iteration, and

$$E_F = \sum_j (q_{x,j} - p_{x,k_j})^2 + (q_{y,j} - p_{y,k_j})^2. \qquad (7)$$

The system optimizes a weighted sum of E_F, E_I, and a regularization term

$$E_P = \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{\gamma_i^2}{\sigma_{M,i}^2} + \sum_i \frac{(\rho_i - \bar{\rho}_i)^2}{\sigma_{R,i}^2}. \qquad (8)$$

that penalizes solutions with low prior probability, based on the standard derivations σ of individual shape, texture, and mouth shape estimated by PCA. ρ_i denotes the rendering parameters, $\bar{\rho}_i$ their starting values, and $\sigma_{R,i}$ are ad-hoc estimates of their standard deviations. The weight of E_I is set to zero in the first iterations, and increased subsequently, while the weight of E_F is decreased and vanishes at the end. Figure 5 shows intermediate states of the fitting procedure. Fitting 99 principal components for individual shape and texture and 10 components for expressions takes 5 minutes on a 2GHz Pentium 4 processor.

After optimization, the linear combination (5) provides estimated albedoes for the entire surface. To capture details such as scars or the strokes of the painter's brush, we perform an illumination-corrected texture extraction [5] on all texture elements visible in the image: Inverting the effect of the estimated illumination, the albedoes of each point on the 3D surface are computed from the pixel values in the image. Weighted by the angle between the surface normal and the viewing direction, these values replace the previous estimate. If several images are available, contributions are automatically pasted into one texture.

Figure 6 shows novel mouth shapes and expressions generated automatically from images and a few feature point coordinates. If the face is not neutral in the input image (first row), our algorithm automatically estimates its neutral shape.

3.2. Neutralization of Faces

The mouth in the reconstructed face can be closed, and a neutral expression can be enforced, by setting the coefficients $\gamma_i = 0$ in Equation 5 after the fitting process (Figure 7). The 3D face is then a linear combination of neutral faces only. In general, removing facial expressions in an image is more difficult than adding new expressions, as it is necessary to be able to reproduce the wrinkles and their shading very precisely. Residual differences between the original and the synthesized image will be falsely attributed to texture in the texture extraction algorithm. Therefore, our algorithm does not yet remove strong or unusual wrinkles completely.

3.3. Background Continuation in Still Images

Near the contour of a face, regions of the background that are occluded in the original image may be revealed as the mouth moves. We therefore replace part of the face in the original image, continuing across the facial contour all structures adjacent to the face. The animated 3D face is then rendered in front of that modified background. The optimal strategy

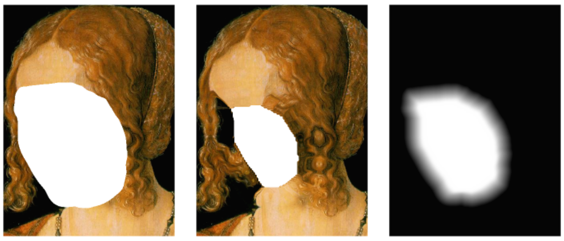


Figure 8: *Based on a segmentation into face and non-face regions (left) provided by the fitting algorithm, the background texture is reflected beyond the contour (center) to avoid artefacts in animation (see Section 3.3).*

for background continuation depends on the background's structure. In our examples, it is important to retain the overall structure of texture of the background, which may for example be a strand of hair (Figure 8). We therefore cannot use Image Inpainting algorithms[3]. Pure texture completion, on the other hand, would require a uniform texture.

For background continuation, our system can rely on a segmentation into face area and background (Figure 8, left) from fitting the morphable model. For a stripe along the contour just outside of the face region, our algorithm reflects all pixel values to the inside, using a smooth warp field. This method retains texture, while keeping discontinuities low. The width of the stripe is calculated from the camera parameters and corresponds to 15mm in the 3D scene.

To compute the warp field, we use an iterative propagation algorithm to calculate the distance $d(x,y)$ from the boundary for all pixels (x,y) within a stripe along the contour (Fig. 8, right). Then, the normalized gradient of the distance map

$$\widehat{\mathbf{g}} = \frac{\mathbf{g}}{\|\mathbf{g}\|}, \quad \mathbf{g} = \left(\frac{\partial d}{\partial x}, \frac{\partial d}{\partial y}\right)^T \tag{9}$$

defines a warp field that reflects points across the edge:

$$(\Delta x(x,y), \Delta y(x,y)) = -2d(x,y) \cdot (g_x(x,y), g_y(x,y)).$$

4. Animating Moving Faces in Video

One of the main benefits of the 3D model as opposed to example-based methods in 2D is the versatility with respect to changes in head pose and illumination. These changes naturally occur in video sequences. In this section, we address the problem of making a person in a given video sequence say a novel text, regardless of what he or she said in the original footage, and retaining the original head movements.

Reanimating video involves the following steps:

1. Recover a textured 3D model from original video frames (Section 3.) If the video contains no large in-depth rotations, it is sufficient to build the face from the first frame only. Otherwise, precision of 3D shape can be increased and texture details from all sides can be included by fitting the model to two or three frames simultaneously.
2. Track 3D head motion (Section 4.1).
3. Generate a trajectory in the coefficients of mouth configurations from audio or text, for example by simple keyframe interpolation.
4. Add the mouth configuration vector to the neutral 3D model at each frame.
5. Render the modified shape on top of the original video frame, using the pose and illumination parameters recovered by the tracking algorithm.

In the reanimated video, part of the background may be revealed that was occluded in the original sequence. We can identify those pixels by comparing the z-buffer values from the reanimated 3D face with those from the tracked face. Unlike still images, the required background region may be visible in previous or subsequent video frames: If the desired pixel is found within 10 frames in either direction, its color value is copied to the current frame. Otherwise, the algorithm uses the method described in Section 3.3.

Figure 9 shows 4 frames from a video recorded at 30fps with a web-cam (640x480 pixels). The video includes large rotations, a non-uniform background, and speech. For 3D shape estimation, we used frame 0, 44, and 66 (out of 150), showing the front, the left and the right side of the face. We labeled 11, 15, and 17 feature points, respectively. No 3D scan of the person was involved in any processing step.

4.1. Tracking

The rigid motion and mouth movements in the input video can be tracked with a method similar to the 3D reconstruction algorithm described in Section 3.1: The algorithm fits the morphable face model to consecutive frames by minimizing image difference E_I (Equation 6) and a regularization term E_P (8) [5, 30, 28]. In each fitting process, the starting values, and the minimum of E_P, are set to the previous frame's result, respectively. Keeping the person's individual shape and texture fixed, we only optimize for rigid transformation and mouth movements (coefficients γ_i of the 4 most relevant principal components, Equation 5). The feature point method that was presented in Section 3.1 is not involved in this process. Since all rendering parameters are estimated from the first frame, no calibration is required. Reliability of the algorithm has been increased significantly by a coarse-to-fine strategy that starts with fitting a down-sampled version of the frame, and then proceeds to full resolution. Computation time is 16s per frame on a 2GHz Pentium 4.

4.2. Speech Synthesis

In a set of demo videos, we show visual speech synthesis from audio signals and text, reanimating faces in a digital photograph, in paintings, and in a video (Figure 9). Phonetic alignment of speech and text has been provided by the CMU-SPHINX system[19], producing a temporal sequence of

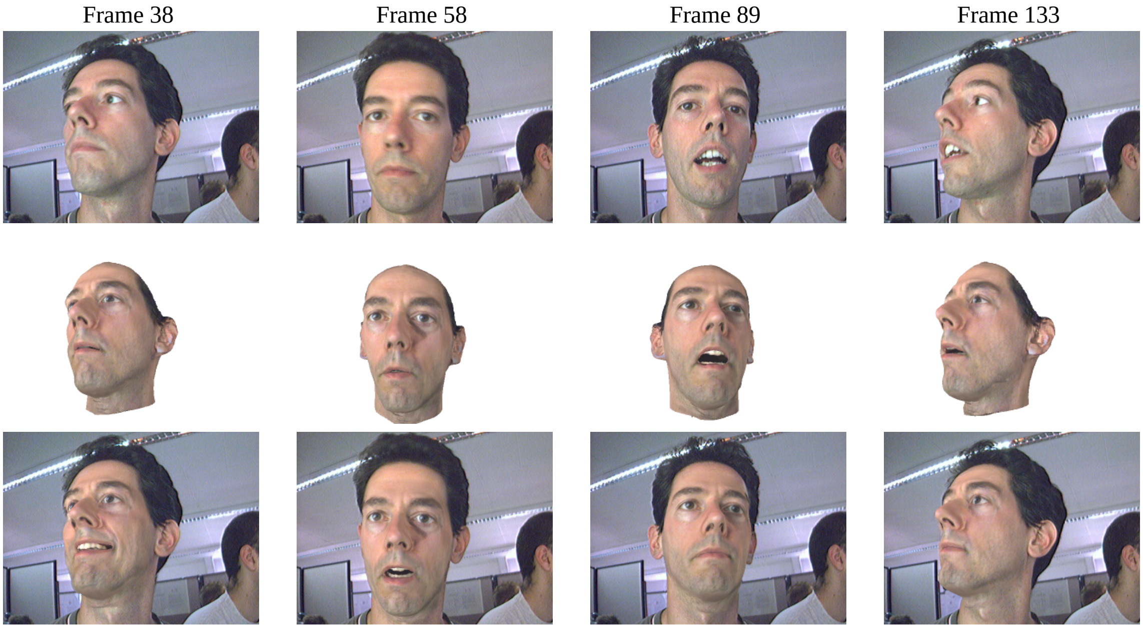


Figure 9: *From each original frame of a video (top row), an estimate of pose and mouth shape was calculated (second row). 3D shape and texture were reconstructed from 3 selected frames. In the third row, the face with new mouth shapes is rendered into the original image.*

phonemes that can be mapped to the visemes of our dataset. Based on this sequence, we perform keyframe interpolation with cosine-shaped acceleration and deceleration. Temporal super-sampling by a factor of 4 is used to produce motion-blur.

5. Conclusions

We have presented a unified method to learn a model of facial expressions and individual neutral faces from 3D scans, and we described a set of algorithms that apply this information to animate a given face in an image or a video. The system is suited for a wide range of applications, due to the low requirements to the input data.

Our framework is open for various future developments: The variation in the expressions of different persons can be investigated in a database of examples. To compensate differences in head shape and size, we could include methods such as Expression Cloning [24], where the direction and length of shape deformations are adapted to the local geometry at each vertex. It is straightforward to include additional expressions recorded from a trained expert in order to cover the entire expressiveness of human faces. Our current database is focused on mouth movements, which seem to be the most challenging problem in facial animation.

As real-time 3D scanning devices are becoming more and more available, 3D snapshots and time-sequences can replace the static scans forming our vector space. With these data, we can use learning techniques to study the dynamics of speech and expressions, and to include coarticulation effects [11]. Since we model speech as a trajectory in a vector space of mouth shapes, more sophisticated dynamic patterns can be implemented easily. Finally, we plan to investigate the non-linear structure of the manifold of faces within our morphable face model by higher-order statistics.

Acknowledgements

We would like to thank Heinrich Bülthoff (Max-Planck-Institute for Biological Cybernetics, Tübingen), for making the laser scanner available, and Barbara Knappmeyer for lending her face. We would also like to thank Tony Ezzat for many interesting discussions, and Cynthia Findlay, who is the speaker in our video. Part of the work was funded by NTT, Japan.

References

1. J.R. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center Princeton NJ 08540, 1990.

2. P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements. In *Advanced Computer Animation, SIGGRAPH '85 Tutorials*, volume 2, pages 61–79, New York, 1985. ACM.

3. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester.

Image inpainting. In *Computer Graphics Proceedings SIGGRAPH 2000*, pages 417–424, 2000.

4. A. Bjorck and G. Golub. Numerical methods for computing angles between linear subspaces. *Math. Comp.*, 27, pp. 579–594, 1973.

5. V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Computer Graphics Proc. SIGGRAPH'99*, pages 187–194, Los Angeles, 1999.

6. M. Brand. Voice puppetry. In *Computer Graphics Proc. SIGGRAPH'99*, pages 21–28, Los Angeles, 1999.

7. C. Bregler, M. Covell, and M. Slaney. Video rewrite:driving visual speech with audio. In *Computer Graphics Proc. SIGGRAPH'97*, pages 67–74, 1997.

8. D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int. J. of Comp. Vision*, 38, 2:99–127, 2000.

9. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2nd edition, 2001.

10. P. Eisert, S. Chaudhuri, and B. Girod. Speech driven synthesis of talking head sequences. In *Proc. 3D Image Analysis and Synthesis '97 – Erlangen*, infix, Sankt Augustin, 1997.

11. T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. In *Comp. Graph. Proc. SIGGRAPH'02*, pages 388–398, San Antonio, 2002.

12. T. Ezzat and T. Poggio. Visual speech synthesis by morphing visemes. *International Journal of Computer Vision*, 38, 1:45–57, 2000.

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

14. S.B. Gokturk, J.Y. Bouguet, and R. Grzeszczuk. A data driven model for monocular face tracking. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, pages 701–708, 2001.

15. T. Goto, S. Kshirsagar, and N. Magnenat-Thalmann. Automatic face cloning and animation. *IEEE Signal Processing Magazine*, 18, 3:17–25, 2001.

16. H. P. Graf, E. Cosatto, and T. Ezzat. Face analysis for the synthesis of photo-realistic talking heads. In *Proc. of the 4th Int. Conf. on Automatic Face and Gesture Recognition*, pages 189–194, 2000.

17. R.M. Haralick and L.G. Shapiro. *Computer and robot vision*, vol 2. Addison-Wesley, Reading, Ma, 1992.

18. T. S. Huang and L. A. Tang. 3d face modeling and its applications. *Int. J. Pattern Recog. Artif. Intell.*, 10(5):491–519, 1996.

19. X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The SPHINX-II speech recognition system: an overview (http://sourceforge.net/projects/cmusphinx/). *Comp. Speech and Language*, 7, 2:137–148, 1993.

20. K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: Generating animated head models with anatomical structure. In *Proc. ACM SIGGRAPH Symp. on Comp. Anim. (SCA) 2002*, pages 55–64, 2002.

21. G. Kalberer and L. Van Gool. Face animation based on observed 3d speech dynamics. In *Procs. of the 14. IEEE Conf. on Comp. Animation (CA'01)*, pages 20–27. IEEE Comp. Society, 2001.

22. R. M. Koch, M. H. Gross, and A. A. Bosshard. Emotion editing using finite elements. In *Comp. Graphics Forum, Vol. 17, No. 3 EUROGRAPHICS '98*, pages C295–C302, Lisbon, Portugal, 1998.

23. Y.C. Lee, D. Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *SIGGRAPH '95 Conf. Proc.*, pages 55–62, Los Angels, 1995. ACM.

24. J. Noh and U. Neumann. Expression cloning. In *Computer Graphics Proc. SIGGRAPH 2001*, pages 277–288. ACM Press, 2001.

25. F.I. Parke. Computer generated animation of faces. In *ACM National Conference*. ACM, November 1972.

26. F.I. Parke. *A Parametric Model of Human Faces*. PhD thesis, University of Utah, Salt Lake City, 1974.

27. F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 75–84, 1998.

28. F. Pighin, R. Szeliski, and D. Salesin. Modeling and animating realistic faces from images. *International Journal of Computer Vision*, 50, 2:143–169, 2002.

29. S. Platt and N. Badler. Animating facial expression. *Computer Graphics*, 15(3):245–252, 1981.

30. L. Reveret and I. Essa. Visual coding and tracking of speech related facial motion. GVU Center Tech Report GIT-GVU-TR-01-16, Georgia Tech, 2001.

31. D. Terzopoulos and Keith Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6):569–579, 1993.

32. T. Vetter and V. Blanz. Estimating coloured 3d face models from single images: An example based approach. In *Computer Vision – ECCV'98 Vol. II*, 1998. Springer, Lecture Notes in Comp. Science 1407.

33. K. Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics*, 22(4):17–24, 1987.