EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Example-Based Rendering of Eye Movements

Michael Banf and Volker Blanz

Institute for Vision and Graphics, Universität Siegen, Germany

Abstract

This paper describes a model for example-based, photo-realistic rendering of eye movements in 3D facial animation. Based on 3D scans of a face with different gaze directions, the model captures the motion of the eyeball along with the deformation of the eyelids and the surrounding skin. These deformations are represented in a 3D morphable model.

Unlike the standard procedure in facial animation, the eyeball is not modeled as a rotating 3D sphere located behind the skin surface. Instead, the visible region of the eyeball is part of a continuous face mesh, and displacements of the iris as well as occlusions by the lids are modeled in a texture mapping approach. The algorithm avoids artifacts that are widely encountered in 3D facial animation, and it presents a new concept of handling occlusions and discontinuities in morphing algorithms.

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

1. Introduction

Parametric models, physics-based approaches and example-based methods, which are the three main paradigms in facial animation, have used different strategies to represent and animate human eyes: Parametric models [Par74], which are most widely used in movie production studios today, and physics-based models [LTW95, KHYS02, SNF05] use 3D models of the eyeballs that are manually designed, based on anatomical data, and that can be rotated to simulate eye movements. Very sophisticated anatomical eyeball models, including detailed iris structures and refraction effects, have been presented by [LBS*03, FGBB02]. In all these models, the eyeballs are located behind a 3D mesh that represents the facial skin and that has openings between the eyelids. Even though these models reproduce the facial anatomy, they have a number of drawbacks: their appearance is often different from that of the surrounding skin, which may have details, noise and blur from 3D scans or photos. The transition between skin mesh and eyeball has sharp edges which pop out visually because they do not fit the spatial frequency spectrum of the face. Also, these models are tedious to construct manually. Finally, it may be difficult to couple the rotation of the eyeball with deformations of the eyelid and the rest of the face.

The main idea of example-based models is to avoid any explicit, manual design of anatomical structures, and to represent and measure only whatever is visible in the face. Ideally, all relevant information about eye movements would be learned from the data set of input data, and this would

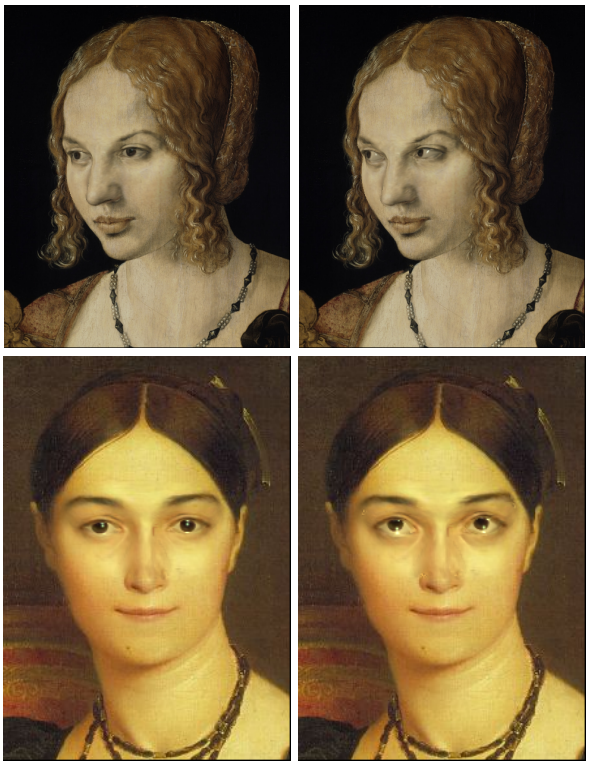

Figure 1: *The eye region of the reconstructed 3D face is rendered into the original paintings (Dürer, Ingres) to reproduce the original gaze (left) or simulate a new gaze (right).*

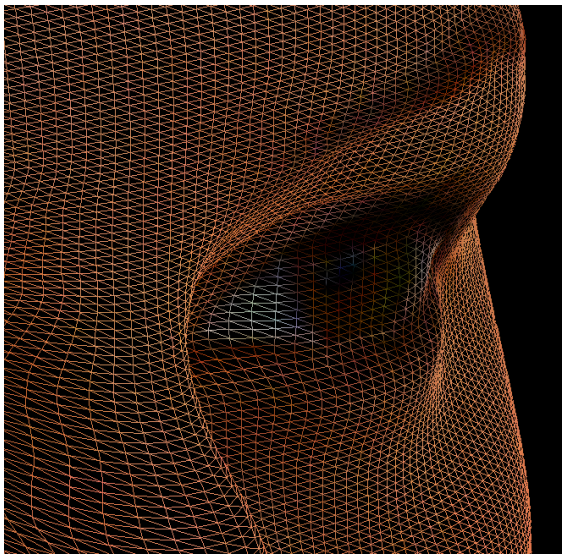

Figure 2: *The continuous face mesh of the morphable model represents the eye just as any other part of the surface. The proposed algorithm uses this mesh, but applies additional texture operations on the eyeball region.*

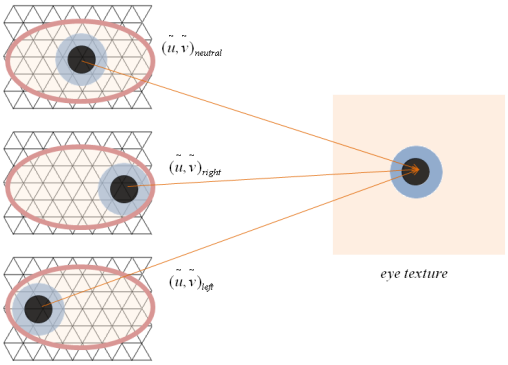

Figure 3: *Each gaze direction i (left column) is assigned its own set of texture coordinates $\tilde{u}_{k,i}, \tilde{v}_{k,i}$ such that the vertex k_i that represents the pupil in this scan is texture-mapped with the pupil texel, while the texture on all other vertices k is shifted along the mesh without distortions.*

be reproduced automatically by the system. Therefore, neither 2D systems [BCS97, EGP02, HWT*04] nor 3D systems [PHL*98, BBPV03, ZSCS04, WHsL*04, VBPP05] include any specific treatment of the eye regions, and they still capture a certain degree of eye movements automatically, depending on the eye movements in the training set. Their focus tends to be on speech synthesis and facial expressions rather than eye movements.

While the focus of our work is on rendering, there have been a number of studies on the behavioural aspect of eye movements: Based on neurobiological findings and image analysis, [IDP03, PI06] predict the eye movements and fixation points of a character when watching video clips. Other approaches learn temporal patterns of gaze and blinking from tracking data and apply statistical methods [LBB02] or methods from texture synthesis [DLN05] to extrapolate these data temporally to generate infinite animation sequences. For the purpose of computing an environment map from the reflections on the eye, a precise anatomical model of the cornea and an algorithm for detecting the edge of the iris have been presented [NN04].

Occlusions of structures in 3D, which occur at the eyes and the mouth, are notoriously difficult to represent in many example-based rendering approaches which, in general, rely on image space (2D methods) or a continuous surface representation (3D methods). This is a result of the fact that they deliberately avoid to consider anatomical details on the design level. For the mouth, occlusions are handled in 2D methods by sophisticated blending schemes from the closest example frames [EGP02], and in 3D methods they have been addressed by dividing the mesh of an open mouth along the inner line of the lips, and letting the lips occlude the teeth and tongue in 3D as the face is morphed towards a closed mouth shape [BBPV03]. However, a similar strategy would fail for the eyes for reasons that we will discuss below.

In this paper, we present a new way of learning the appearance of eye movements from 3D scans, and rendering these movements in a photo-realistic way in a 3D morphable model framework [BV99]. The main idea is that we capture the deformation of the eyelids and the surrounding tissue, which we observe in sample scans, by the standard 3D morphable model approach with a uniform 3D mesh representation [BV99, BBPV03], while the visible region of the eyeball is *part of the same, continuous surface mesh*, yet with a different texture (Figure 2). The texture coordinates are shifted as the eye moves (Figure 3), and the face is morphed at the same time. The transition between eyelid and eyeball is handled by alpha blending in texture space.

The novel theoretical contributions of this approach are:

- Example-based modeling of eye movements,
- A new texture map scheme that interpolates texture coordinates rather than texture color or vertex positions,
- Alpha blending in texture space,
- Usage of a single, continuous mesh (Figure 2) for the entire face that still handles occlusions.

Practical benefits for facial animation are:

- Coupled movements of eyeball and the surrounding skin,
- Uniform appearance of eyeball and skin in terms of tone, blur and noise,
- Avoiding the sharp transition between skin and eyelids that makes many existing face models look unnatural,
- Easy, mostly automated construction of new face models, as opposed to anatomical models of eyeballs,
- Transfer to new individuals based on the morphable model.

2. Eyes in a 3D Morphable Model

The main idea of 3D morphable models [BV99] is to represent different samples of faces (individuals or expressions) by the same 3D face mesh with a fixed set of vertices. Each vertex represents a given structure in the face consistently across all samples. Building a 3D morphable model, therefore, involves solving the *correspondence problem*: for each vertex on a reference scan, the corresponding surface points on all sample scans need to be identified. This can be done by optical flow [BV99] or by model-based fitting [BSS07] of an existing morphable model to the data.

After concatenating the 3D positions and the red, green and blue color values of all $n = 75972$ vertices into shape and texture vectors

$$\mathbf{S}_i = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T \quad (1)$$

$$\mathbf{T}_i = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T, \quad (2)$$

we can form linear combinations of samples

$$\mathbf{S} = \sum_{i=1}^{m} a_i \mathbf{S}_i, \quad \mathbf{T} = \sum_{i=1}^{m} b_i \mathbf{T}_i \quad (3)$$

with real-valued coefficients a_i and b_i in a meaningful way to represent new faces.

We expanded the morphable model of [BV99] by taking into account high-resolution textures (800×800 texels.) For a given vertex k, we use the same texture coordinates u_k, v_k for all faces, so all textures are in correspondence. The texture coordinates are equivalent to the cylindrical coordinates $u_k = \phi_k$, $v_k = h_k$ of the reference face of the morphable model. These textures form a vector space just as the texture vectors $\mathbf{T}_i$ defined above (Equation 2 and 3).

For synthesizing new gaze directions on the eyelids and the surrounding facial surface, we use this vector space property of the morphable model to perform bilinear interpolation between the shape vectors of the 4 nearest neighbour directions:

$$\mathbf{S} = \sum_{i=1}^{4} a_i \mathbf{S}_i, \quad \sum_{i=1}^{4} a_i = 1, \quad (4)$$

while we use the texture of the frontal gaze scan throughout the animation. With a correct correspondence (see section 3), this will give valid results. The eyelashes are mapped on the eyelids in this texture, which is a good approximation in near-frontal head poses and gaze directions.

In contrast, the eyeballs need special treatment due to the fact that they are occluded to different degrees by the eyelids: The morphable model always consists of the same, continuous surface mesh (Figure 2) that is registered with all visible points in the example scans. For structures that are occluded in a scan, such as the pupil in a closed-eye scan, it is unclear where to map it, while other structures above or below the iris may appear that are not represented by the reference mesh.

In the following paragraphs, we discuss four options how the eyes could be represented in the morphable model, and we argue why Option 4 gives the best visual results:

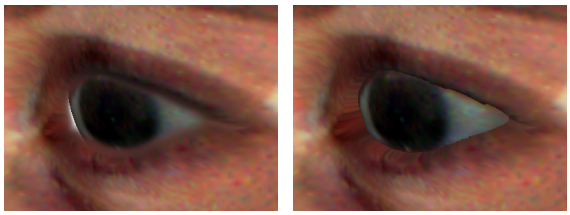

Figure 4: *If the eyeballs are modeled as separate spheres behind openings in the face mesh, gaps between the meshes (left, slightly reduced by alpha blending) are visible as the face is morphed. Extending the triangles of the eyelid edge backwards (right) does not look satifactory either. Also, the sharp transition between face mesh and eyeball does not fit to the smooth appearance of the face (right image, right half of the eye).*

2.1. Option 1: A Rotating Sphere

We could devote an additional set of vertices and triangles to the eyes. These could be, as in parametric or physics-based models, a textured sphere which rotates in 3D space. For each gaze direction i, the vertices of the eyeball have 3D positions that are stored in $\mathbf{S}_i$. Then, linear interpolation (Equation 4) will, in general, not define a rotated version of the sphere, but a distorted or shrunk version due to the non-linear nature of the rotation. Still, it would be feasible to implement a generalized morphing function with special treatment of the eyeballs in terms of a 3D rotation. However, fundamental disadvantages of this approach are the fact that there will be a sharp transition between the triangles of the eyelids and those of the eyeballs where the meshes touch or intersect, and these form salient, sharp features in potentially blurred faces. Moreover, it is difficult to maintain a waterproof mesh structure as the eyelids and surrounding tissue morph during the animation. This results in gaps or in distorted triangles along the eyelids (Figure 4).

2.2. Option 2: Interpolation of Textures

A simple morph or cross-dissolve between eye textures (Equation 3) is clearly insufficient, because the iris will fade out in one position and fade in on another position as the gaze direction moves from one sample direction to another.

2.3. Option 3: Interpolation of 3D Vertex Coordinates

Ideally, the morphable model would take care of the gaze directions automatically due to the correspondence, and the eye would just be part of the continuous surface mesh (Figure 2): If it is always the same vertex $k \in \{1, \ldots, n\}$ that is located for example on the center of the pupil or on the left edge of the iris in all gaze samples $\mathbf{S}_i$, $\mathbf{T}_i$, then the gaze directions would be captured by the vertex displacements across faces, and the texture would be the same throughout

the dataset. For example, the texture values R_k, G_k, B_k of the pupil vertex would be black in all $\mathbf{T}_i$.

However, as the eyeballs slide under the eyelids, triangles would be compressed, while others would be extended where the eyeball slides off the eyelid and new regions of the iris or sclera appear. These mesh deformations along the edge of the visible region of the eyeball may be acceptable for small eye movement only.

2.4. Option 4: Interpolation of Texture Coordinates

The method that we propose in this paper represents the eyes by the vertices of the surface mesh (Figure 2), but with a separate texture map and an additional set of texture coordinates $\tilde{u}_k, \tilde{v}_k$. This texture map is generated from one of the sample scans or 3D face reconstructions, and it is kept constant throughout the animation. Unlike Option 3, the motion of the vertices does not depend on the gaze direction. As the eyelids close, they converge to a single line in 3D, and they expand when the eyes are opened. The texture slides over these vertices due to a change in texture coordinates (Figure 3). The texture coordinates $\tilde{u}_k$, $\tilde{v}_k$ of vertex k are calculated by a bilinear interpolation between those of the 4 closest sample gaze directions:

$$\tilde{u}_k = \sum_{i=1}^{4} a_i \tilde{u}_{k,i}, \quad \tilde{v}_k = \sum_{i=1}^{4} a_i \tilde{v}_{k,i}, \tag{5}$$

with the same coefficients a_i that are used for shape deformation of the eyelids and surrounding tissue (4). As a result, each vertex obtains the color value of the eyeball point that it represents in a given intermediate gaze direction. This defines a morphable model of texture coordinates.

To create a smooth transition, we define an alpha mask with a transition in opacity along the inner edges of the eyelids, and let the skin and eyeball textures overlap slightly along that edge. Occlusions of the eyeball by the eyelids are captured in a natural way by this approach.

In the next section, we describe how we establish correspondence and preprocess the scans. Section 4 describes how the texture map of the eyeball region is created, and how it is shifted by interpolating texture coordinates.

3. Preprocessing the 3D Scans

The training database is a set of 11 3D scans of a person: nine have gaze directions that vary in the horizontal and vertical direction (Figure 5), one has both eyes closed, and one has one eye closed (winking). To make the correspondence problem between scans easier to solve, we placed color dots on the face as markers (these are not essential for the algorithm.) The data were recorded in a frontal pose with a structured light phase-shift scanner *VitroScan-3D v3* in a sampling grid of approximately *0.5mm* in a horizontal and vertical direction. Between 0 and 106 (median 12) vertices out of 82 000 had to be filled by simple interpolation, due to holes caused by specular reflections, eyebrows and eyelashes. No

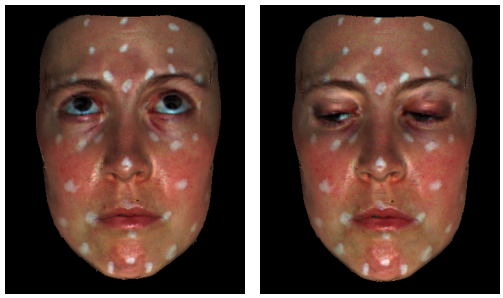

Figure 5: *Structured light scans are used as a training set for learning the geometric changes that occur during eye movements. The set includes 9 gaze directions, a scan with one eye closed (winking) and one with both eyes closed.*

Figure 6: *The eyelid lines are Catmull Rom splines along 6 manually defined points on each eye. They are used to refine the correspondence.*

holes due to self-occlusion occured on the inner part of the face. The quality of the scans is sufficient to capture details such as the bulge of the upper eyelid and the fold above the lid.

We first establish correspondence between the frontal gaze scan and our previously developed *morphable model of identities* [BV99] of 200 individuals at a frontal gaze direction. Then, we establish correspondence between the frontal gaze scan and all other gaze directions. Combined, all eye movements can be mapped back to the morphable model of identities.

3.1. Correspondence

Correspondence between scans at frontal gaze directions can be established by fitting the morphable model of identities to 3D scans using the iterative algorithm described in [BSS07]. This algorithm finds the linear combination (3) that is closest to the scan in shape and texture, along with the optimal rigid transformation and illumination. It is initialized by manually clicking 5 points on the face (the tip of the nose and the corners of the eyes and mouth). As a result, we can identify each vertex of the morphable model of identities on the scan surface, and store the correct (scanned) positions and colors as new values in the shape and texture vectors $\mathbf{S}_i$, $\mathbf{T}_i$. The 3D coordinates are automatically aligned by a rigid transformation.

For precise correspondence along the eyelids, we have to refine this because the morphable model of identities has no degrees of freedom that would compensate the movements of eyelids or eyeballs.

This procedure starts by manually clicking 6 points for

Figure 7: *The texture (800×800 texels) extracted from a 3D scan or an image using the 3D morphable model, is parameterized in cylindrical coordinates. Additional eye textures are inserted in the corners. In the face, the skin color is interpolated automatically into the eyes and will be blended with the eyeball textures along the eyelids.*

each eye along the inner edge of the eyelids on the eyeball. Catmull-Rom splines along these points define the eyelid lines (Figure 6). The scans are now represented by six arrays in a cylindrical parameterization:

$$x(h,\phi), y(h,\phi), z(h,\phi), r(h,\phi), g(h,\phi), b(h,\phi), \qquad (6)$$

where (h,ϕ) are the coordinates of the corresponding point on the reference face, so they serve as surface parameters only and have no geometrical interpretation here. We warp the arrays (6) such that the eyelid curves are matched. On the eyeball, the warp field is interpolated from the curves, compressing or expanding it vertically in the (h,ϕ) parameterization as the eyes open or close. The pupils and irises are not considered here, so they will *not* be in correspondence. Outside of the curves, the displacement decreases linearly over a range of approximately 30mm. Note that while this warp along the (h,ϕ) parameterization establishes correspondence between scans, it leaves the 3D surface geometry of each scan unchanged.

3.2. Definition of Regions

Based on the eyelid lines, we define a mask for the eyeball region of the mesh, which is used for separate texture mapping in Section 4. This mask is defined on the reference face, but it is valid for all gaze directions due to the precise correspondence. A blurred version of this mask serves as an alpha mask for blending. Also, we apply different specularity parameters in the Phong illumination model on the eyeballs to simulate the glossy surface on the eyeballs and along the inner edge of the eyelids.

3.3. Spherical Eyeballs

Due to the optical properties of the lens and pupil, scanners tend to produce errors on the iris and pupil which have to be corrected. We do this by fitting a sphere to all vertices of the eyeball in each scan, and displacing the vertices in depth towards the reconstructed sphere. When fitting the sphere, we assume that the radius is 11mm, which is an anatomical parameter that is relatively constant across individuals, and that in the center of mass of the vertices of the eyeball, the tangent plane is frontoparallel, so the center of the sphere is located behind it. Therefore, only the depth value of the center has to be estimated by a least squares procedure. By modeling the surface of the eyeball as a sphere, we ignore the fact that the lens produtes slightly, while the iris is in fact located inside of the sphere.

Figure 8: *Mapping a regular grid texture on the eyeball demonstrates the undistorted rendering of the eye texture in frontal views.*

4. Texture-Mapping the Eyeball Region

This section describes how to create a texture of the visible region of the eyeball that can be used for all of the gaze directions, and how to assign texture coordinates $\tilde{u}_{k,i}$, $\tilde{v}_{k,i}$, to each vertex k for each gaze direction i.

Note that the vertices do not, in general, form a regular mesh for each gaze i: the mesh may be compressed, expanded or distorted due to the correspondence of the eyelids. As the eyeball moves, the texture will slide over the surface due to the changes in texture coordinates, and these changes have to compensate this distortion such that the texture remains undistorted and the iris is circular in each rendered image, i.e. for all interpolated vertex positions (Equation 4) and texture coordinates (Equation 5).

We achieve this by inverting the distortion of the mesh in each individual sample gaze, and using an undistorted eyeball texture throughout the interpolation. The algorithm is based on the following two assumptions: (a) In frontal gaze directions, the iris is circular when projected orthographically on the frontoparallel (x,y) plane. (b) The rotation of the eyeball can be approximated by a shift of the texture along the frontoparallel plane (i.e. left-right and up-down). Both assumptions make sense because the visible part of the eyeball covers a small angle of the entire sphere only.

For each vertex k and each gaze direction i, located in $(x_{k,i}, y_{k,i}, z_{k,i})$ in 3D space, we set

$$\tilde{u}_{k,i} = s \cdot x_{k,i} + t_{u,i}, \quad \tilde{v}_{k,i} = s \cdot y_{k,i} + t_{v,i}, \qquad (7)$$

with a scaling factor s that guarantees appropriate texture resolution, and a translation $t_{u,i}, t_{v,i}$ that will be discussed below. In our implementation, we embed the eyeball textures in the corners of the overall face texture (Figure 7).

With these texture coordinates, we can map a regular grid texture on each scan to verify that it appears undistorted in the image plane (Figure 8).

4.1. Generating the Eyeball Textures

With the texture coordinates (7), we can generate a texture from any face **S**, **T** of the morphable model. This may be a scan of a face with the eyes wide open, but also a 3D reconstruction of a face from a single image, computed with the algorithm [BV99] using high-resolution textures. The texture generation is implemented as a rasterization from the 3D model into the texture space, using $\tilde{u}, \tilde{v}$ (corners in Figure 7).

The texture of the eyeball captures only whatever is visible in the scan or input image, unlike traditional approaches that represent the entire eyeball. The rationale behind our approach is that regions that are invisible in images or scans of wide open eyes will be invisible in any other gaze direction. Still, it is possible to combine segments from different images or scans into one texture due to correspondence, or to extend the sclera and iris manually or automatically as in traditional approaches. In some cases, it may also be necessary to remove specular highlights from the textures. We did a slight manual extension and removed highlights for the examples shown in this paper.

4.2. Motion of the Eyeball Texture

In order to make sure that the simultaneous interpolation of vertex and texture coordinates produces the desired motion of the iris along the surface, we set the translation variables $t_{u,i}, t_{v,i}$ such that the vertex k_i that lies on the center of the pupil in scan i will be mapped to the center of the pupil of the texture (Figure 3). We achieve this by manually selecting the center of the pupil in each sample gaze and in the reference face of the morphable model of individuals.

The algorithm will map the center of the pupil always to the desired point on the mesh surface, move the rest of the texture according to a constant translation (which approximates rotation), and maintain undistorted appearance throughout the interpolations, as we demonstrate in the next paragraph.

4.3. Verification of the Undistorted Eyeball Appearance

From Equations (5) and (7), we obtain for each vertex k

$$\tilde{u}_k = \sum_{i=1}^{4} a_i \tilde{u}_{k,i} = \sum_{i=1}^{4} a_i (s \cdot x_{k,i} + t_{u,i}) \tag{8}$$

$$= s \sum_{i=1}^{4} a_i x_{k,i} + \sum_{i=1}^{4} a_i t_{u,i} = s x_k + \sum_{i=1}^{4} a_i t_{u,i} \tag{9}$$

because we form linear combinations for x_k and $\tilde{u}_k$ with the same coefficients a_i. The same holds for $\tilde{v}_k$ and y_k. In a frontal pose and orthographic projection into an image, the pixel position (x'_k, y'_k) of the vertex (x_k, y_k, z_k) is

$$x'_k = s' \cdot x_k + t_x, \quad y'_k = s' \cdot y_k + t_y \tag{10}$$

with a scaling factor s' and a translation t_x, t_y. Substituting x_k from (9) into (10), we see that the relationship between texture coordinates and pixel coordinates is linear:

$$x'_k = \frac{s'}{s} \tilde{u}_k - \frac{s'}{s} \sum_{i=1}^{4} a_i t_{u,i} + t_x, \tag{11}$$

$$y'_k = \frac{s'}{s} \tilde{v}_k - \frac{s'}{s} \sum_{i=1}^{4} a_i t_{v,i} + t_y. \tag{12}$$

This projection will render the texture in isotropic scaling and with a constant shift that depends on the weights a_i, i.e. on the gaze direction. More specifically, the iris remains circular, which makes sense for the small natural rotation angles of the eyeball. Because the texture is mapped on the spherical eyeball geometry, we obtain a natural appearance of the iris as the face rotates and perspective projection is applied.

4.4. Closed eyes

For closed eyes, the shape vector $\mathbf{S}_i$ morphs the eyelids together and compresses the eyeball polygons to a single line as the upper and lower eyelid curves (Section 3.1) become identical. To make sure that the iris does not move when the eyes close, we use Equation (7) with the same translation constants $t_{u,i}, t_{v,i}$ as in the desired gaze direction (e.g. frontal gaze).

5. Rendering the Face Model

In addition to the standard rendering pipeline, the algorithm involves the following steps:

1. For an intermediate gaze direction, find the four closest neighbour samples. Then, the gaze angles in these samples and the target gaze define the linear weights a_i (bilinear interpolation according to horizontal and vertical angle).
2. Form a linear interpolation of vertex coordinates (4).
3. Form a linear interpolation of texture coordinates (5).
4. Use the constant high-resolution texture of face and eyeball.
5. During rasterization, perform two texture lookups per fragment, with texture coordinates u, v and $\tilde{u}, \tilde{v}$, respectively.
6. Blend the contributions of both textures according to the alpha mask that is defined in texture space.
7. Use a texture map that defines the specular parameters of the phong model in each vertex to make the eyes more glossy than the skin.

We are using a software rendering implementation of this algorithm, but it is straight-forward to implement it on a GPU.

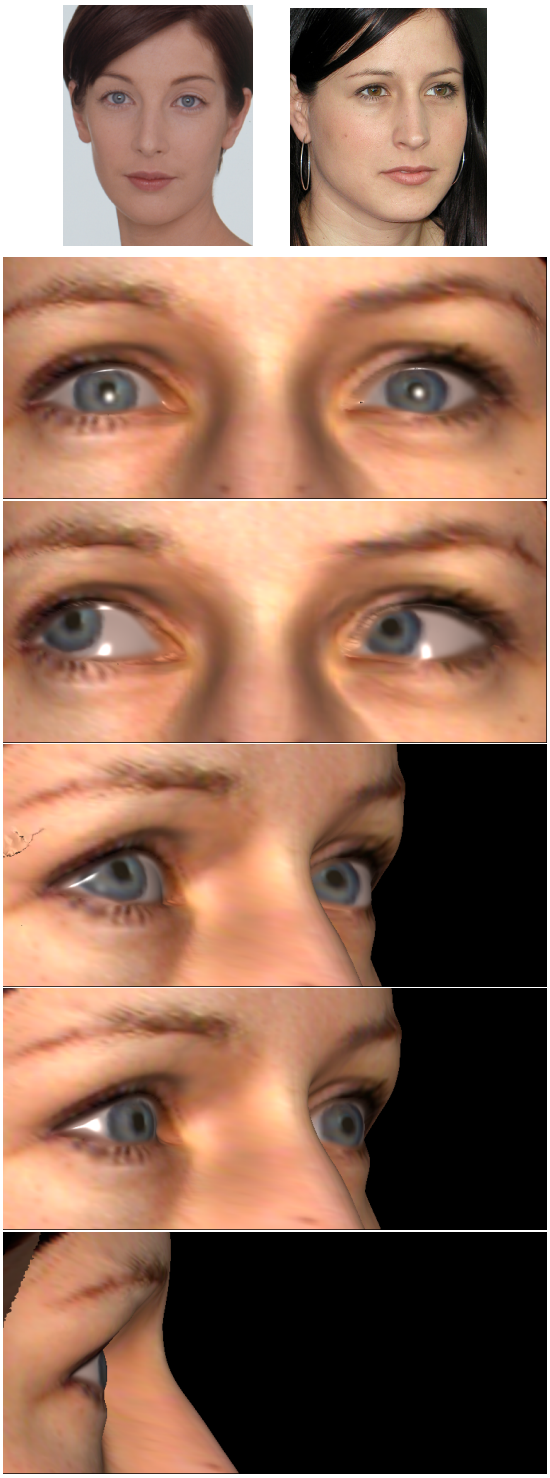

Figure 9: *After reconstructing 3D faces from the two input images (top), the eyes of the first person are inserted into the face of the second by our algorithm.*

Figure 10: *Closing the eyes of the reconstructed face model (Figure 9) by our algorithm.*

5.1. Transfer to New Faces

Due to the morphable model, it is straight forward to transfer eyes and eye movements between individuals: We simply exchange the eye texture, compute new texture coordinates for the new face (7), form difference vectors $\mathbf{S}_{gaze} - \mathbf{S}_{neutral}$, $\tilde{u}_{k,gaze} - \tilde{u}_{k,neutral}$ and $\tilde{v}_{k,gaze} - \tilde{v}_{k,neutral}$ from the sample face and add these to the new face.

Figures 9 and 10 show how the algorithm inserts new eye textures (in this case from a photograph) into a 3D face. Embedded in a system similar to [BBPV03], we can animate faces in photos and paintings (Figures 11 and 1). The video shows more examples of faces. The eyes have white specular reflections on the eyeball and along the lids.

6. Conclusions

We have presented a general solution for adding photorealistic eye movements to morphing-based facial animation. The main challenge is the fact that the eyelids occlude different parts of the eyeball as the eyes move. Our method avoids a number of discontinuity artifacts that are common in parametric and physics-based models, and it presents a new paradigm of using textures on 3D meshes. Manual interaction is reduced to clicking a small number of points on each sample scan, and perhaps some postprocessing of the eye texture.

In our model, the eyelashes are part of the texture, so they give correct results only in near-frontal poses and gaze directions (Figures 9 and 10), and they do not cast shadows

Figure 11: *New eye textures are inserted (top, right) into the original face (top left) and animated (bottom row).*

on the eyeball, unlike the eyelids. Another limitation ist that it is not straight-forward to account for refraction of light on the cornea, as it has been done in some of the anatomical 3D models of eyeballs [LBS*03, FGBB02]. In closeup side views, the transparent lens should be visible. However, methods from image-based rendering may help to solve this problem effectively in the future. Pupil dilation would be easy to simulate with our texture coordinate morphing, even more so than in existing 3D approaches. Our new morphing paradigm proves to be efficient and precise: It scales well with increasing resolution of the model, and it makes it easy to obtain and transfer eyeball textures from photographes.

References

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating faces in images and video. In *Computer Graphics Forum, Vol. 22, No. 3 EUROGRAPHICS 2003* (Granada, Spain, 2003), Brunet P., Fellner D., (Eds.), pp. 641–650.

[BCS97] BREGLER C., COVELL M., SLANEY M.: Video rewrite:driving visual speech with audio. In *Computer Graphics Proc. SIGGRAPH'97* (1997), pp. 67–74.

[BSS07] BLANZ V., SCHERBAUM K., SEIDEL H.-P.: Fitting a morphable model to 3d scans of faces. In *IEEE International Conference on Computer Vision* (2007).

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *Computer Graphics Proc. SIGGRAPH'99* (1999), pp. 187–194.

[DLN05] DENG Z., LEWIS J. P., NEUMANN U.: Automated eye motion using texture synthesis. *IEEE Comput. Graph. Appl. 25*, 2 (2005), 24–30.

[EGP02] EZZAT T., GEIGER G., POGGIO T.: Trainable videorealistic speech animation. In *Computer Graphics Proc. SIGGRAPH'02* (San Antonio, 2002), pp. 388–398.

[FGBB02] FRANÇOIS G., GAUTRON P., BRETON G., BOUATOUCH K.: *Anatomically accurate modeling and rendering of the human eye*. Rapport de recherche isnr inria/rr, INRIA, 2002.

[HWT*04] HAWKINS T., WENGER A., TCHOU C., GARDNER A., GORANSSON F., DEBEVEC P.: Animatable facial reflectance fields. In *2004 Eurographics Symposium on Rendering* (2004), Keller A., Jensen H. W., (Eds.), pp. 309–319.

[IDP03] ITTI L., DHAVALE N., PIGHIN F.: Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Proc. SPIE 48th Annual International Symposium on Optical Science and Technology* (2003), vol. Vol. 5200, SPIE Press, pp. 64–78.

[KHYS02] KÄHLER K., HABER J., YAMAUCHI H., SEIDEL H.-P.: Head shop: Generating animated head models with anatomical structure. In *Proc. ACM SIGGRAPH Symposium on Comp. Anim. (SCA) 2002* (2002), pp. 55–64.

[LBB02] LEE S. P., BADLER J. B., BADLER N. I.: Eyes alive. *ACM Trans. Graph. 21*, 3 (2002), 637–644.

[LBS*03] LEFOHN A., BUDGE B., SHIRLEY P., CARUSO R., REINHARD E.: An ocularist's approach to human iris synthesis. *IEEE Comput. Graph. Appl. 23*, 6 (2003), 70–75.

[LTW95] LEE Y., TERZOPOULOS D., WATERS K.: Realistic modeling for facial animation. In *SIGGRAPH '95 Conference Proceedings* (Los Angels, 1995), ACM, pp. 55–62.

[NN04] NISHINO K., NAYAR S. K.: Eyes for relighting. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 704–711.

[Par74] PARKE F.: *A Parametric Model of Human Faces*. PhD thesis, University of Utah, Salt Lake City, 1974.

[PHL*98] PIGHIN F., HECKER J., LISCHINSKI D., SZELISKI R., SALESIN D.: Synthesizing realistic facial expressions from photographs. In *Computer Graphics Proceedings SIGGRAPH'98* (1998), pp. 75–84.

[PI06] PETERS R. J., ITTI L.: Computational mechanisms for gaze direction in interactive visual environments. In *ETRA '06: Proceedings of the 2006 symposium on Eye tracking research & applications* (New York, NY, USA, 2006), ACM, pp. 27–32.

[SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (2005), 417–425.

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIC J.: Face transfer with multilinear models. In *Computer Graphics Proc. SIGGRAPH'05* (2005), pp. 426–433.

[WHsL*04] WANG Y., HUANG X., SU LEE C., ZHANG S., LI Z., SAMARAS D., METAXAS D., ELGAMMAL A., HUANG P.: High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *In EUROGRAPHICS* (2004), pp. 677–686.

[ZSCS04] ZHANG L., SNAVELY N., CURLESS B., SEITZ S. M.: Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 548–558.