Eurographics Symposium on Rendering (2004)
A. Keller, H. W. Jensen (Editors)

Animatable Facial Reflectance Fields

Tim Hawkins[1], Andreas Wenger[1], Chris Tchou[1], Andrew Gardner[1], Fredrik Göransson[2], and Paul Debevec[1]

[1] University of Southern California Institute for Creative Technologies, United States
[2] Linköping University Norrköping Visualization and Interaction Studio, Sweden

Abstract

We present a technique for creating an animatable image-based appearance model of a human face, able to capture appearance variation over changing facial expression, head pose, view direction, and lighting condition. Our capture process makes use of a specialized lighting apparatus designed to rapidly illuminate the subject sequentially from many different directions in just a few seconds. For each pose, the subject remains still while six video cameras capture their appearance under each of the directions of lighting. We repeat this process for approximately 60 different poses, capturing different expressions, visemes, head poses, and eye positions. The images for each of the poses and camera views are registered to each other semi-automatically with the help of fiducial markers. The result is a model which can be rendered realistically under any linear blend of the captured poses and under any desired lighting condition by warping, scaling, and blending data from the original images. Finally, we show how to drive the model with performance capture data, where the pose is not necessarily a linear combination of the original captured poses.

1. Introduction and Background

Creating realistic, animatable human facial models has been a longstanding endeavor in computer graphics. The central aspects of the problem have been characterizing facial geometry, characterizing facial reflectance properties, and characterizing how the face should move. Capturing facial geometry is now common practice, with commercial products such as Cyberware scanners able to record millimeter-accurate data of a face. Facial geometry has visually important structure at finer scales as well, which [HEG01] models and renders.

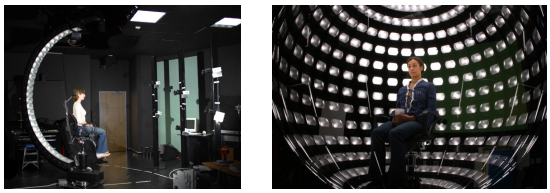
Capturing facial reflectance is important for being able to render faces realistically, especially for rendering faces under novel illumination conditions. [MWL*99] measured a composite BRDF for human skin using a small number of images under different lighting conditions, leveraging the presence of a variety of surface normals observable within each image. [MGR00] extrapolated this BRDF to a complete 3D face model and rendered performance-captured animations of the result. [HK93] made the important observation that human skin exhibits subsurface scattering, which has prevented skin rendering using BRDFs only from achieving realistic results. Faster computing and improved algorithms for simulating subsurface scattering [JMLH01] now make rendering translucent materials such as skin practical.

Image-based approaches to facial modeling, rendering, and animation have shown that photorealistic results can be obtained without specific facial reflectance models, and in some cases without geometric models as well. [BN92] showed that photorealistic faces in intermediate expressions and poses could be created through image-based morphing. [BCS97] resequenced video to match annotated speech, using morphing to smooth the lip motions. [EGP02] employed a multidimensional morphable model to synthesize new mouth animation from a relatively small set of example images. Image-based techniques that make use of geometric models enable increased control over the rendered viewpoint. Pighin [PHL*98] captured different expressions and registered them to a generic 3D model, using view-dependent texture mapping and expression blending as in [Par74] to synthesize new expressions from new viewpoints. All of these techniques render the face in the same illumination as the original images. [BV99] used 3D scans and images to create a morphable model representing a subspace of possible individuals in neutral expressions. Their model could be rendered under variable illumination, but used a relatively simple reflectance and illumination model. [BBPV03] have extended this morphable model to include

variations in facial expression. [DHT*00] used an image-based technique to render human faces in arbitrary lighting environments by recombining a large set of basis images from different lighting directions, and used a geometric model to render the faces from new viewpoints. [MPN*02] captured similar datasets from a much larger set of viewpoints and demonstrated photoreal renders from arbitrary viewpoint and under arbitrary illumination, but due to the long acquisition times this technique is limited to inanimate objects. In our work, we use a lighting apparatus similar those of [DHT*00] and [MPN*02], but optimized for rapid capture of human subjects. We capture the subject in a variety of expressions from several directions to allow changes in expression and animation as well as viewpoint and illumination.

Generating facial motion can be done through simulating the physics of facial structure, musculature, and tissues as in [TW90, LTW95]. Alternately, performance-driven animation [Wil90], uses a real human performance to drive the motion of a computer animated face. This approach was used in [GGW*98], which projected face images tracked from six cameras to render performances from novel viewpoints; [MGR00] also generated facial motion in this manner. While most of these techniques track points marked on the face, [PSS99] tracks and renders performances by fitting a morphable model to facial motion sequence. [BBPV03] likewise track performances using a morphable model. Recent work at ESC Entertainment for the sequels to *The Matrix* used video performance capture not only to generate motion, but also to reproduce changes in skin albedo during performance. Like these works, we do not synthesize novel motions, but instead track motion from real performances to drive our face model. In this way, our work is also similar to [BFJ*00], which rendered captured facial performance data by blending a small number of hand-drawn cartoons of characters in different facial expressions. In our work, we use a similar approach to blend our photoreal facial datasets to match a captured performance.

2. Data Capture



(a) (b)

Figure 1: *(a)The lighting apparatus and camera setup. (b) An eight second long-exposure photograph of the strobes firing. A total of 480 incident illumination directions are captured.*

A key component in our setup is the lighting apparatus [HCD01] shown in Figure 1. This device consists of a semi-circular arm along which are arrayed 30 strobe lights. By rotating the arm around the vertical axis and flashing the strobes in sequence, a subject seated in the center can be illuminated from several hundred directions in just a few seconds. Synchronized video cameras placed outside of the arc of the arm capture one frame for each lighting direction.

Our system uses six Sony DXC-9000 cameras equipped with zoom-lenses, which run at 60 frames per second. Each of the 6 cameras is connected to a PC equipped with a capture card.

2.1. Capture session

We first place fiducial dots on the subject's face and neck using a flourescent yellow "gel-ink" pen. Such pens produce dots that are non-toxic, easy to apply, easy to remove, and very bright and easy to detect. We automatically remove the dots from our image data before rendering; the dots were kept as small as possible to facilitate this.

After applying the dots, the subject sits in the chair at the center of the lighting apparatus and adopts a comfortable posture and head pose. The subject is then asked to assume various facial expressions, mouth deformations, head poses, and eye positions, and for each of these a dataset is captured. For each dataset, the subject remains still for eight seconds while the arm rotates, the strobes flash, and the cameras capture frames. Each dataset comprises 480 images corresponding to 480 different lighting directions.

Each computer computes and displays the sum of a representative sampling of the captured frames. If the image is blured, indicating subject motion, the capture is repeated. Verifying and saving the data takes about 20 seconds, and while the data saves the subject prepares for the next expression to be captured. It is thus possible to capture 60 expressions in only 30 minutes.

After all expressions have been captured, the positions and intrinsic parameters of the video cameras are calibrated using the technique presented in [Zha00]. The intensity response characteristics of the cameras are calibrated by capturing a lighting dataset of a grayscale chart.

2.2. Choosing expressions

We were particularly interested in capturing many different mouth shapes, since the mouth has an extremely wide variety of possible shapes. We chose shapes that would capture the different deformations typically found in each region of the face. Because we can capture an expression quickly relative to the overhead involved in preparing for a data capture session, we chose to capture a fairly large number of expressions for each subject.

One example of a captured set of expressions is shown in Figure 3. We chose to include:

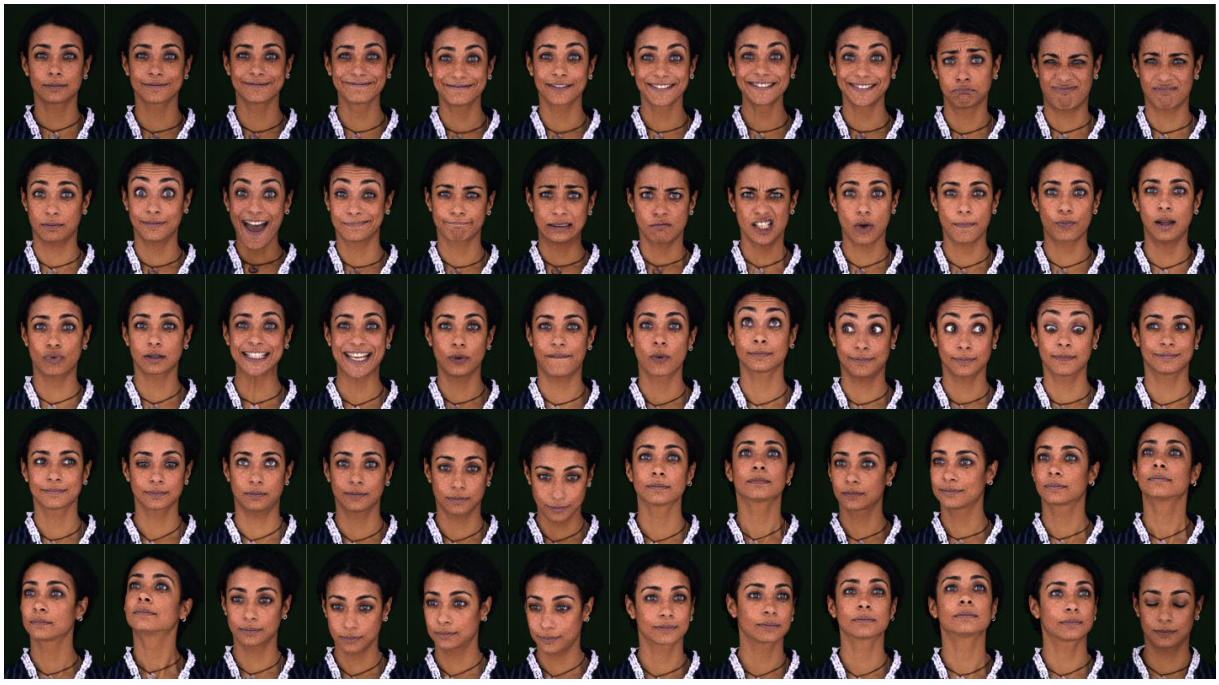


Figure 3: *The set of captured expressions, including a variety of emotional expressions, visemes, head poses, and eye poses, as well as reference coverage of areas such as teeth and eyelids.*

Figure 2: *The six camera views.*

- Emotional expressions (smile, frown, surprise, disgust, etc.)
- Visemes (/f,v/, /w,r/, /p,b,m/, /t,d,s,z/, /uh/, /oo/, etc.)
- A variety of mouth shapes
- Different head poses
- Different eye positions
- Coverage of seldom seen areas (eyes wide open, eyes closed, teeth bared)

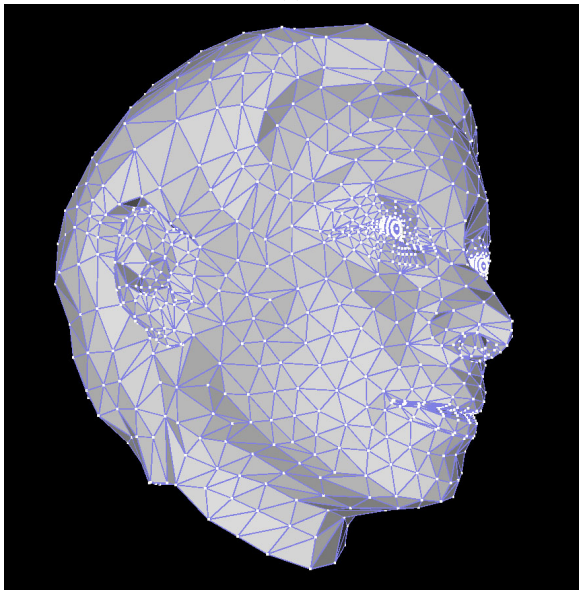
3. Creating the Base Face Model

We require reasonably accurate geometry for all parts of the face and head we would like to render. Because our video cameras are geometrically calibrated, we can triangulate the locations of the fiducial dots from multiple views to obtain their 3D locations. This set of points forms the basis of our 3D face model.

We begin by placing 3D points in the appropriate locations for the neutral expression. This is accomplished using an interactive program, shown in Figure 4, that displays all of the camera views of a given expression, and allows the user to place 3D points in locations corresponding to the observed fiducial locations.

After placing the points, the user specifies edges between the various points, creating a triangle mesh. This mesh contains 300 points, accurate enough for much of the surface of the face. However, the model must have somewhat more detail in areas such as the nostrils and ears, and must also extend properly to coincide with the edges of the lips and the eyelids. A model for the upper and lower teeth is also needed. For each of these areas, a good reference expression was chosen which shows as much of the area as possible. Additional vertices were then added to the face model and

(a)



(b)

Figure 4: *(a)The interactive point-editing and registration program. The user can select an expression and verify the registration from all six camera views simultaneously, and can adjust misregistered points. (b) The face mesh.*

placed to align well with the six camera views of the reference expression. The results of this modeling are shown in Figure 4. The model for the teeth is very simple, consisting of two triangle strips, one for the upper teeth and one for the lower teeth, with two triangles per tooth. The total user time for building this initial model was about six hours–one hour for specifying the 3D locations and connectivity of the fiducials, and five hours for modelling extensions (hair, teeth, eyes) and additional detail (nostrils, eyelids).

4. Registering the expressions

Building a morphable model from our datasets requires placing the image geometries of each dataset into correspondence with the 3D face model. For most of the face, identifying the locations of the fiducials in 3D space provides sufficient information. However, the fiducials give no information about the precise location of the teeth or the eye/iris, and these must be handled specially. Also, high contrast boundaries such as the eyelid/eyeball boundary and the lip/mouth boundary must be registered very precisely and are also handled specially.

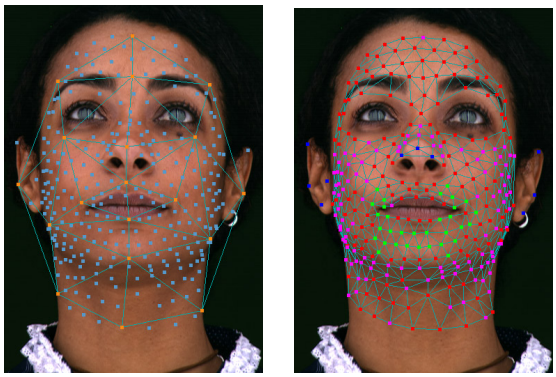
The registration process leverages our ability to use our reflectance datasets to compute lighting-independent image proxies for the different expressions. For example, when detecting the yellow fiducials, we use images that are lit by completely diffuse lighting, except omitting a cone of lighting directions opposite from the camera, which largely eliminates bright glancing specularities. Because the virtual lighting for these images is different for each camera (omits a different cone of lights based on the camera direction), the resulting images could not be produced by any single static lighting environment on the subject. Similarly, when marking the irises and eyelid boundaries of the eyes we use images where each pixel represents the 95th percentile of brightness (over the varying lighting directions), which effectively removes the specularities from the eyeball and minimizes shadowing near the eyelid boundary, making these features much easier to identify.

4.1. Dot registration

To minimize potential texture misalignment, we chose to use a relatively large number of dots as fiducials for registering images of the face in different expressions and from different cameras. Using 300 dots, and capturing 60 expressions from six different cameras, we arrive at a total of 108,000 2D dot locations that must be determined (neglecting occlusion). We estimate that using a simple GUI/mouse interface to manually mark this number of points would take 100 hours of (extremely tedious) work. We developed a 3D registration technique that automates most of this task.

First, we manually place a small subset of 3D points (18 points) for all expressions, using the 3D interface described in Section 3. This takes about one minute per expression. Using the neutral expression, a coarse triangle mesh connecting the sparse set of points is specified.

A dense registration of all the expressions is then computed automatically. The registration algorithm expands the number of densely registered expressions one at a time, using the set of expressions already densely registered. We start with only the neutral expression, which was densely registered in Section 3. For each new expression N, we achieve a dense registration by first computing an initial estimate of each 3D point position using the coarse meshes, then



(a) (b)

Figure 5: *(a) The coarse mesh is used to automatically fill in the remaining dots. (b) The chosen dimensionality for the local deformation space of each vertex. Blue=0, magenta=1, red=2, green= 3. The 3-dimensional points are concentrated near the mouth.*

applying a progressive 3D snap-to algorithm which tries to place the 3D points so that they lie on yellow dots in as many views as possible.

For each point p, we use a distance-weighted version of the deformation metric described in Section 5 to determine which expression R among the already fully registered expressions is most similar to N in the neighborhood of p. We compute an initial estimate of p's position in the new expression by computing a 3D morph of p's position in R. This morph is computed using the positions of corresponding triangles of the coarse mesh in R and N, analogous to the 2D morph presented in [BN92].

After computing an initial position estimate for all points of the new expression, we apply a progressive 3D snap-to algorithm:

1. Initialize the set S of already-snapped points to contain the known 3D points of the rough mesh
2. Initialize the predicted 3D offset of each point in S to be the difference between its morph-prediction and its actual known position
3. Initialize the dot-strength threshold to be the strongest dot response of the known points
4. For each point p bordering S
   a. Compute a predicted offset by averaging all of the offsets of points in S connected to p
   b. Starting from the predicted offset, perform a 3D-snap operation
   c. If the dot-strength of the snapped position is above the dot-strength threshold, set the position of p, add p to S, and store the offset of p from its original morph-prediction.
5. If some new points were added to S, repeat Step 4.
6. Lower the dot-strength threshold. If it remains above a specified fraction of the initial threshold, go to step 4.

By starting with a high threshold for adding points to S and gradually lowering it, propagation of errors due to weakly observed or invisibly dots is avoided.

The 3D-snap operation itself simply searches in a small 3D neighborhood to find the location that yields the best dot-strength. Dot-strength is computed by projecting the 3D point into each of the camera view images and summing the response of the resulting 2D points to a linear filter tuned to detect the yellow fiducial dots. For each point, its visibility from each camera view is computed using the initial estimate of the face mesh, and views from which the point is occluded are excluded when computing the total filter response.

4.2. Registration of eyes, teeth, lips

In addition to registering the dots, we must accurately register areas that had no dots, such as eyes and teeth. For the eyes, we mark the irises from multiple views. This gives the location of the front of the eye. For expressions where the eye is believed to be pointing at the center camera, we place the center of the eye accordingly. From these expressions, we determine the average location of the centers of the eyes relative to the head pose. The centers of the eyes are then set to these same relative positions for expressions where the eyes are not pointing toward the center camera.

The upper and lower teeth are rigid bodies; we just need to find a translation and rotation for each of them for each expression. Because the upper teeth are rigidly attached to the skull, we simply place them in the same location relative to the head pose. We estimate head pose by using one point on each ear and one point on the bridge of the nose, which together determine a rigid transformation. The position of the lower teeth is approximated using a single marked 3D point, which determines their translation relative to the reference expression, and the relative rotation is approximated as the head pose rotation.

Our technique for placing the lips is somewhat more complicated. We rotoscope the lip boundary in at least two camera views, and then initialize the lip geometry using a set of fiducial dots near the lips to morph the lip geometry from the reference expression to the new expression. This estimate is then iteratively relaxed to make its silhouette align with the rotoscoped boundary in each of the rotoscoped view. The eyelids are rotoscoped and warped into place in the same way. This rotoscoping is performed for each of the expressions; the total time to rotoscope the lips and eyelids in all of the expressions was about three hours.

We remove the dots from the reflectance field data using a technique similar to that of [GGW*98], and our results use dot-free reflectance fields.

4.3. Rendering blends of expressions

The result of the 3D registration process is a morphable model similar to that obtained by [PHL*98], with the added

benefit that the model can be rendered with any desired illumination condition. Some renders of linear combinations of the captured expressions and views are shown in Figure 9, with varying illumination.

5. Blend Weights for Arbitrary Deformations

To create a fully animatable model, we would like to be able to blend the source data appropriately to match any arbitary target deformation resulting from motion capture or other facial animation techniques. We wish to render each triangle of the target mesh by blending source images that best represent the appearance of that triangle under the target deformation. The primary assumption we make is that the important changes in appearance are a function of facial geometry. This assumption can account for changes in shadowing due to wrinkles as well as changes in skin albedo due to stretching and compression. Obviously, this assumption cannot account for effects such as changes in blood content due to emotion alone (which need not involve any facial deformation at all).

For most of the face we make the additional assumption that changes in appearance are a function of *local* deformation. Because we do not explicitly model shadowing effects (they are implicit in the reflectance field), this assumption is only valid for areas of the face where the non-local shadowing can be approximated as constant with respect to changes in facial geometry. For instance, a point on the cheek can be shadowed by the nose, but because of the limited mobility of the nose its exact deformation has very little effect on shading of the point on the cheek. The assumption does not hold for the teeth and eyes, where the lighting is very dependent on the exact positions of the lips and the eyelids. In the cases of the teeth and eyes, we don't use expression blending at all, instead rendering from a single expression (in which they were maximally visible) and using ray-tracing to explicitly correct for shadows. However, for the continuous surface of facial skin, we found that the assumption of local dependence is a reasonable one.

When computing expression blends, we consider only vertices of the face mesh corresponding to fiducials, since their locations are exactly known. Our analysis is performed independently for each vertex v, and proceeds as follows.

We consider the set of vertices V with shortest path from v less than or equal to two. Because we wish to consider only changes in local deformation, we apply a rigid transformation R_i^S to the positions $P_i^S(V)$ for each of the source expressions S_i, $i = 1,...,n$. Each rigid transformation R_i^S is chosen to minimize the sum of squared distances between the positions $R_i^S(P_i^S(V))$ and the positions of the vertex set in the neutral expression, $P_0^S(V)$. We computed the R_i^S using the closed-form solution presented in [Hor87].

To determine the source expressions to blend together for vertex v in an arbitrary target expression T_j, we first compute R_j^T. We know the source deformation vectors $R_i^S(P_i^S(V))$ for each of the source expressions $i = 1,...,n$, as well as the deformation vector $R_j^T(P_j^T(V)$ for the target expression T_j. Determining a good blend of source expressions for rendering v in expression T_j can then be cast as a scattered data interpolation problem. Our solution is based on [BFJ*00]

We first perform principal components analysis (PCA) on the set of deformation vectors $R_i^S(P_i^S(V))$, $i = 1,...,n$. We then choose a dimensionality m such that the sum square of the projections of all the deformation vectors onto the $n-m$ smallest eigenvectors falls below a threshold. In practice we limit m to be no greater than 3, to simplify subsequent computations. The chosen dimensions are shown in Figure 5(b).

We then compute an m-dimensional Delaunay triangulation of the projections of the $R_i^S(P_i^S(V))$. For the target deformation, we project its deformation vector and compute which m-dimensional cell C of the Delaunay simplex it lies in. The source expressions corresponding to the vertices of this cell are then used as the desired blend expressions for the vertex v in the target shape T_j, with blend weights equal to the barycentric coordinates of the projected target deformation vector relative to C.

5.1. Coherent expression culling

We expect the set of captured source expressions to contain considerable redundancy. For example, we capture many expressions with different mouth configurations. For most of these, the subject's forehead will be very close to its resting deformation, providing no new information for rendering that area. To reduce the amount of source data required for rendering, as well as to be able to blend and interpolate smoothly between expressions without introducing temporal scintillation artifacts, we would like to remove as much of this redundancy as possible.

We perform expression culling for each of several user-defined regions, including the forehead, mouth, neck, throat, chin, and left and right cheeks and eyes. For each region, sets of source expressions are chosen that satisfy a maximal independent set criterion, i.e. that adding any other expressions to the set would cause there to be two expressions with deformations that are too similar. Consistent with our measure of local deformation near a vertex, we rigidly transform the vertex positions in a region for a given expression to a least-squares match with the neutral expression. The resulting vertex positions comprise the deformation vector for that region and expression. Connecting expressions where the L2 norm of the difference in the deformation vectors falls below a threshold gives a classic maximum independent set problem. This problem is NP-complete; we compute an approximate solution using the Minimum Degree Greedy algorithm.

We cull entire regions at once rather than perform a separate culling for each vertex in order to minimize the number of separate datasets used in each local region. Our maximal

independent set algorithm could easily yield very different expression sets for adjacent vertices even if the deformation distances between all expressions are similar for the two vertices.

6. Rendering Relightable 3D Morphable Models

The capture and registration process provides us with a relightable 3D morphable model consisting of the geometry for each of the source expressions and sets of texture coordinates that map each reflectance field onto the geometry. Given this model, a set of expression blendweights for each vertex (computed as in Section 5 or with any other desired method), and a desired novel camera position and illumination condition, the task of the renderer is to sample from the reflectance fields to generate a novel image of the face with the specified expression blend, viewpoint, and illumination.

We wish to synthesize a lit texture for the target geometry by using a small set of source expressions and a small set of views for each vertex. Blending the appropriate weights yields the local reflectance field for the neighborhood of the vertex under consideration. Those reflectance fields are then blended across the triangles to provide the reflectance function at every point on the geometry. The final step is to apply a lighting environment to the reflectance function at every point on the geometry to synthesize the lit texture.

In the remainder of this section we will describe the process of picking the best set of cameras to sample from, how to blend between the expression view combination and how we accumulate and light all the weighted reflectance fields to render the final image.

6.1. Sampling Viewing Directions

For a given vertex and source expression to be rendered, we sample from the six viewpoints based on two factors: First, the position of the new virtual viewpoint relative to the original camera positions, and second, the local pose of the novel expression relative to the local pose of the source expression.

For the viewpoint sampling we use the inverse look vector of the camera in polar coordinates. Our cameras are mapped into this space and Delaunay triangulated. Sampling from the capture cameras simply involves finding the triangle the novel view point falls into and calculating the according barycentric weights.

Assuming the pose in the novel expression is identical with the poses of the source expressions we sample from, the above method would be correct. However, if there is a difference in pose between the novel expression and the source expressions we have to compensate for each of the pose differences individually. The reflectance fields capture the exitant radiance of a point on the face for a given incident illumination direction and exitant direction, where both incident and exitant directions are measured with respect to the local normal. A pose change results in a rotation of this normal, changing the incident and exitant angles. Similar to [WAA*00], we rotate the camera as well as the lighting by the rotation R_{ts} of the local pose of the novel expression t into the local pose of the expression we sample from s. Figure 6 explains this schematically with just a single surface and its normal. To sample the correct capture cameras with the correct blend weights we therefore need to rotate the novel view point by R_{ts} before mapping it into our camera evaluation space.

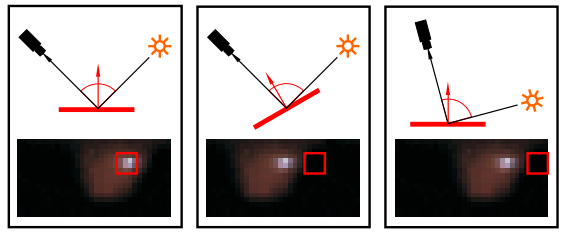


Figure 6: *The left image shows the lighting direction, the viewpoint and the surface orientation as it was in one of the expressions we sample from. This configuration corresponds to an entry in the reflectance function at the point on the surface. The middle image shows a desired configuration of surface patch, camera position, and lighting direction to be rendered. The surface patch has rotated relative to the original, causing a change in the incident and exitant angles with respect to the normal. The change in the reflectance function due to the change in incident angle can be approximated as a rotation of the original reflectance function, shown in the middle at bottom. Equivalently, we can leave the reflectance function in its original form but sample from the location corresponding to the rotated direction, shown at the right bottom. At the right top is shown the rotated light direction, as well as the rotated camera. The camera rotation does not affect the sampling location within a reflectance function, but is used to compute the correct view dependent blending between the reflectance functions captured from the different cameras.*

We compute the local rotations R^i_{ts} by calculating an optimal rigid transform for each vertex i of the face geometry between the novel expression t and the expression we sample from s. The local rigid transform is computed as in Section 5.

Taking the local rotation into account requires us to sample the six cameras differently for each vertex in every expression we sample. We also must apply the inverse of the local rotation when sampling lighting directions from the reflectance field.

6.2. Blending and Lighting the Reflectance Fields

For a given vertex to be rendered, there is a set of source expressions with nonzero blend weights. For each of these source expressions, the viewpoint sampling provides us with a set of one to three capture cameras with their blend weights. Because the viewpoint compensation for local pose depends on the source expression, the camera weights are different for the different source expressions. The camera weighting also varies from vertex to vertex due to the compensation for local pose. When accumulating a given reflectance field corresponding to a source expression and capture camera into the final render, the per vertex weighting for this accumulation is simply the product of the expression blend weight and the camera blend weight. The final reflectivity of a point on the geometry is the accumulation of all the weighted reflectance fields. The blending and accumulation process is illustrated in Figure 7.

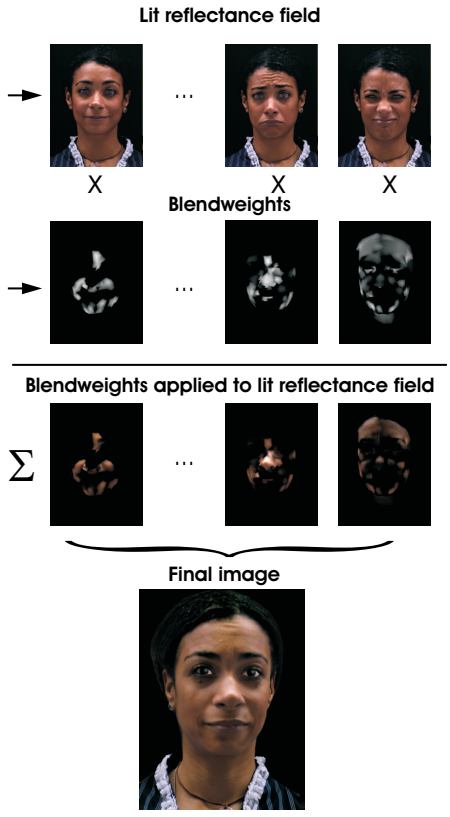


Figure 7: *The top row shows the reflectance fields depicted by a stack of images with different lighting directions. The second row shows the product of the expression and viewpoint blend weights for the different reflectance fields. The third row shows the reflectance field lit by a lighting environment and with the blend weights applied. The final step is to accumulate all these individual weighted and lit reflectance fields to synthesize the final image.*

6.3. Implementation

The implementation of our renderer can be split in two parts. The fist part is responsible for gathering all the information needed to render a reflectance field for a given frame in the animation. This information is stored in a structure we call a render chunk. It holds information such as the frame number, reflectance function ID, geometry, texture coordinates, per vertex lighting rotation, and per vertex blend weights. In this setup part we additionally gather a list of all the reflectance fields needed to render the animation. The second part is to traverse this list of render chunks to render the final image.

The final images are rendered in two stages. First we render all areas not strongly influenced by local moving occluders, that is, the entire face except the eyes and the teeth. We render eyes and teeth in a separate pass because our 60 captured expressions can not adequately sample the rapid variation in shadowing resulting from the motion of the eyelids over the eyes and the lips over the teeth. Instead we render eyes and teeth using only a single reference expression for each (in which they are maximally visible). We then perform a visibility calculation to capture this shadowing. For the eyes, we process the reflectance field to remove the specularity, and then resynthesize the specular reflection in the eyes. This compensates both for the fact that our capture cameras are too widely spaced to capture the view-dependent behavior of sharp specularities, as well as for the fact that these sharp specularities are clamped (saturated) in the original datasets. Both render stages make use of floating point buffers and fragment shaders on modern GPUs. We used ATI's Radeon 9800 XT with 256MB texture memory.

6.4. Tracking

To demonstrate renderings of our model with realistic motion, we captured facial performance motion data of the subject. We did this immediately after capturing the reflectance datasets. The subject was lit as diffusely as possible using several fresnel spotlights and bounce cards. We then captured video sequences at 60 fps using the six cameras while the subject performed. The fiducial dots used during reflectance capture were left in place. Tracking the dots from the different cameras allows us to triangulate 3D motion data.

The eyes are placed using a technique similar to that described in Section 4.2. The irises are again marked, although now we have the option of marking only keyframes and interpolating between them. The eye centers are placed relative to the head pose.

The lips are also placed as in Section 4.2. Again, we need rotoscope only a few keyframes where the shapes of the lips

are changing most dramatically. The in-between frames are computed by applying the motion of the dots nearest the lips.

7. Results

Figure 9 shows several renderings of animatable facial reflectance fields. The top three rows show linear blends of geometry and expression. The lighting environment for each row is shown on the left, and the same three expression blends are used for each of the lighting environments. The second and third rows also demonstrate rendering from novel viewpoints. The bottom row shows three frames from a performance-captured animation rendered using the animatable facial reflectance field. The animation is composited into a background plate corresponding to the lighting environment used to illuminate the face. All of the renderings show subtle changes in the apparent geometry and shading of the face, such as furrowing of the brow, which result from blending the reflectance fields.

The animations accompanying this paper show animated versions of the image sequences in Figure 9. For the blend renderings, the expressions change smoothly, and each intermediate rendering appears realistic. Subtle shifts in the position of specular reflections show the effect of both the view-dependent texture mapping and the rotation of the lighting environment according to changes in head pose. The performance-captured animation shows consistent lighting with the novel background and closely follows the original performance. Though new expression weights are chosen automatically for each facial region for each frame, the face is generally free of sharp changes during the animation.

8. Future Work

The experience with our technique suggests several avenues for future work. First, our rendering process is very data-intensive, and rendering from a compressed representation of our data would desirable. Recent work in representing radiance transfer functions using spherical harmonics [RH02, SKS02] or wavelets [NRH03] could possibly be adapted for this purpose. Second, having a well-motivated method for adjusting for changes in shadows and visibility could allow a significant reduction in the number of expressions that need to be captured. Capturing faces from additional cameras could help reduce tracking problems and allow increased control over the rendered viewpoint. If the back of the head could also be captured, it would be of interest to combine this approach with image-based techniques for capturing hair geometry and reflectance in [GSML02]. Finally, taking greater advantage of emerging hardware rendering techniques based on frequency-space illumination (e.g. [SKS02]) might enable creating renderings of this form in real time.

9. Conclusion

In this paper we presented a technique for creating an animatable and relightable image-based model of a human face. We showed that our model is able to capture appearance variation over changing facial expression, head pose, view direction, and lighting condition, and we have shown how to drive the model with performance capture data, where the pose is not necessarily a linear combination of the original captured poses. We believe the work indicates potential practicality for animation techniques based on relightable face datasets.

10. Acknowledgements

We gratefully acknowledge Marc Brownlow for graphic design work on the figures, Brian Emerson for geometric modelling assistance, Jessi Stumpfel and Andrew Jones for their help with data capture and processing, Shane Chen for his help in making facial masks, and Jessica Vallot for sitting as a subject. Thanks to Frederic Pighin and Jonathan Cohen for their assistance and insightful discussions during an early version of this work. This paper was developed with funds from the U.S. Army Research Institute for the Behavioral and Social Sciences under ARO contract number DAAD 19-99-D-0046. Any opinions, findings and Conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Department of the Army.

References

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating faces in images and video. In *EUROGRAPHICS Annual Conference Proceedings* (2003).

[BCS97] BREGLER C., COVELL M., SLANEY M.: Video rewrite: Driving visual speech with audio. *Proceedings of SIGGRAPH 97* (1997), 353–360.

[BFJ*00] BUCK I., FINKELSTEIN A., JACOBS C., KLEIN A., SALESIN D., SEIMS J., SZELISKI R., TOYAMA K.: Performance-driven hand-drawn animation. In *Proceedings of NPAR* (June 2000), pp. 101–108.

[BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. *Computer Graphics (Proceedings of SIGGRAPH 92) 26*, 2 (1992), 35–42.

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. *Proceedings of SIGGRAPH 99* (August 1999), 187–194.

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. *Proceedings of SIGGRAPH 2000* (July 2000), 145–156.

[EGP02] EZZAT T., GEIGER G., POGGIO T.: Trainable videorealistic speech animation. In *Proceedings of ACM SIGGRAPH 2002* (July 2002), Computer Graphics Proceedings, Annual Conference Series, ACM Press / ACM SIGGRAPH, pp. 388–398.

[GGW*98] GUENTER B., GRIMM C., WOOD D., MALVAR H., PIGHIN F.: Making faces. *Proceedings of SIGGRAPH 98* (1998), 55–66.

[GSML02] GRABLI S., SILLION F., MARSCHNER S. R., LENGYEL J. E.: Image-based hair capture by inverse lighting. In *Proc. Graphics Interface* (May 2002), pp. 51–58.

[HCD01] HAWKINS T., COHEN J., DEBEVEC P.: A photometric approach to digitizing cultural artifacts. In *Proc. 2nd International Symposium on Virtual Reality, Archaeology, and Cultural Heritage (VAST 2001)* (December 2001), pp. 333–342.

[HEG01] HARO A., ESSA I., GUENTER B.: Real-time photo-realistic physically based rendering of fine scale human skin structure. In *Twelfth Eurographics Workshop on Rendering* (June 2001), pp. 53–62.

[HK93] HANRAHAN P., KRUEGER W.: Reflection from layered surfaces due to subsurface scattering. *Proceedings of SIGGRAPH 93* (August 1993), 165–174.

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. In *J. Opt. Soc. Am. A* (April 1987), vol. 4, pp. 629–642.

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proceedings of SIGGRAPH 2001* (August 2001), Computer Graphics Proceedings, Annual Conference Series, ACM Press / ACM SIGGRAPH, pp. 511–518. ISBN 1-58113-292-1.

[LTW95] LEE Y., TERZOPOULOS D., WATERS K.: Realistic modeling for facial animation. *Proceedings of SIGGRAPH 95* (1995), 55–62.

[MGR00] MARSCHNER S., GUENTER B., RAGHUPATHY S.: Modelling and rendering for realistic facial animation. In *Eleventh Eurographics Workshop on Rendering* (June 2000), pp. 231–242.

[MPN*02] MATUSIK W., PFISTER H., NGAN A., BEARDSLEY P., ZIEGLER R., MCMILLAN L.: Image-based 3d photography using opacity hulls. In *Proceedings of ACM SIGGRAPH 2002* (July 2002), Computer Graphics Proceedings, Annual Conference Series, ACM Press / ACM SIGGRAPH, pp. 427–437.

[MWL*99] MARSCHNER S. R., WESTIN S. H., LAFORTUNE E. P. F., TORRANCE K. E., GREENBERG D. P.: Image-based BRDF measurement including human skin. *Eurographics Rendering Workshop 1999* (June 1999).

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (July 2003), 376–381.

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

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

[PSS99] PIGHIN F., SZELISKI R., SALESIN D.: Resynthesizing facial animation through 3d model-based tracking. In *International Conference on Computer Vision* (1999), pp. 143–150.

[RH02] RAMAMOORTHI R., HANRAHAN P.: Frequency space environment map rendering. *ACM Transactions on Graphics 21*, 3 (July 2002), 517–526.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3 (July 2002), 527–536.

[TW90] TERZOPOULOS D., WATERS K.: Physically-based facial modelling, analysis, and animation. *Journal of Visualization and Computer Animation 1*, 2 (August 1990), 73–80.

[WAA*00] WOOD D. N., AZUMA D. I., ALDINGER K., CURLESS B., DUCHAMP T., SALESIN D. H., STUETZLE W.: Surface light fields for 3d photography. *Proceedings of SIGGRAPH 2000* (July 2000), 287–296.

[Wil90] WILLIAMS L.: Performance-driven facial animation. *Computer Graphics (Proceedings of SIGGRAPH 90) 24*, 4 (August 1990), 235–242.

[Zha00] ZHANG Z.: A flexible new technique for camera calibration. In *IEEE Trans. Pattern Anal. Machine Intell.* (2000), vol. 22, pp. 1330–1334.

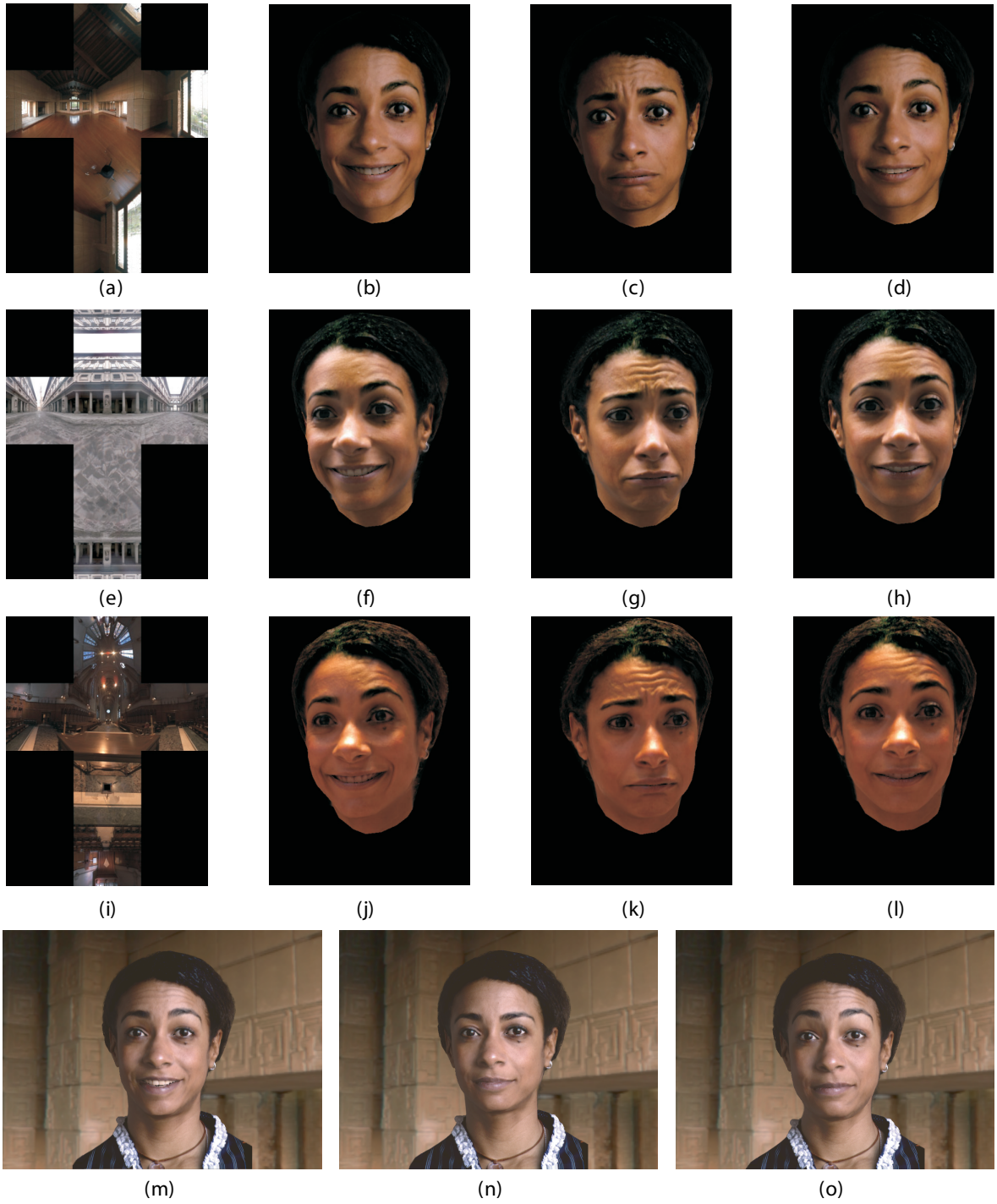


Figure 8: *(a) A captured interior illumination environment. (b-d) Morphed expressions rendered using the illumination in (a). (e,i) Two additional lighting environments. (f-h,j-l) Morphed expressions with novel viewpoints rendered in lighting environments (e,i). (m-o) Frames from a performance-captured animation applied to the face model. The face is illuminated by the environment in (a) and composited into a background image of the corresponding scene.*