Blending Face Details: Synthesizing a Face Using Multiscale Face Models

Seung-Hyun Yoon ■ Dongguk University

John Lewis and Taehyun Rhee ■ Victoria University of Wellington

Creating realistic 3D face models has long been an important and challenging problem in computer graphics. Because humans are sensitive to any facial abnormality, face models in films and games have stringent requirements and often require details such as wrinkles, pores, and more. Recent advances in scanning technology have made it possible to obtain such details. However, this often requires special scanning devices or intensive postprocessing steps. Creating a 3D face model from scratch is far more challenging and is regarded as a highly laborious and time-consuming task, even for a skillful artist.

In entertainment applications such as movies and games, human face models often must be created by artists rather than using an automated method. For example, many characters in animated movies and games are imaginary, so they cannot be scanned. Thus, there is need for a facial design method that is simultaneously simple, understandable, and powerful for artists. We present a system that elegantly addresses this need.

Our starting observation is that the parameterization of the problem should be understandable by artists. A principal component model, although powerful, is not a suitable semantic parameterization, for example. Our approach might be compared to blendshapes, which are a simple but powerful approach that is prevalent in facial animation for generating arbitrary expressions as a weighted blend of a number of basis expressions. The method provides semantic parameterization, which means that the animator can intuitively adjust the strength or influence of the various expressions while blending. This approach has been widely used in film and video game production.1

For our goal of designing faces (rather than facial expressions), one obvious semantic parameterization is a parts-based decomposition, in which a face is designed by selecting the desired eye, nose, and mouth shapes and then blending them together.2 In a complementary approach called natural categorization, the face is hierarchically decomposed into scales of its spatial details, from coarse to fine.3 Researchers have also adopted multiscale face models (MFMs) to represent expression wrinkles in facial animation.4–6

In this article, we present a novel approach to synthesize a new 3D face model using weighted blending of multiscale details across different faces including human (see Figure 1a) and nonhuman characters (see Figures 1b and 1c). Our approach decomposes face models into component scales, with a correspondence of salient facial features across faces. Specifically, we create a MFM that hierarchically represents the face’s spatial details. A 3D face mesh is parameterized by a set of coefficients that encode the expression wrinkles across different face models.
into 2D parameter space and decomposed into a base surface and multiscale continuous displacement maps (CDMs). Each MFM represents face details from coarse to fine scales while providing full correspondences across CDMs.

Our MFMs provide full correspondences across both scales and faces. The salient face regions are marked by 2D feature curves over the parameter space using the interactive user controls. Then, the feature curves drive correspondences across MFMs even for the faces with unusual proportions. Lastly, a new 3D face is interactively synthesized by weighted blending of the MFMs. Using the semantic controls across scales and faces, a user can intuitively synthesize a new face model while blending multiscale details across faces even in highly different shapes (see Figure 1c).

The proposed approach provides considerable power, while providing natural interactions for the artist: the contributions of various scales can be directly specified, and facial features such as salient wrinkles can be interactively marked to enhance the correspondence across faces. Because our system can incorporate new face scans and models as they become available, it becomes more powerful with additional use.

**Background in Realistic 3D Face Models**

Since Frederic Parke presented his pioneering works on face modeling and animation,7 creating and animating a realistic face model has been one of the important and challenging topics in computer graphics, with many associated research publications. Our technique also involves mesh correspondence,8,9 parameterization, 10 and multi-resolution techniques, 11 which are equally important and broad areas. We cannot fully survey these areas, so this section will rely on area surveys and mention only selected examples of alternative approaches.

Thanks to recent advancements in scanning techniques, it is possible to capture high-resolution 3D face geometry. Recent systems can scan face details, such as wrinkles and pores.12–16 As we mentioned earlier, these methods often require a special scanning hardware and sophisticated post-processing steps, such as nonrigid registration and reconstructions.

A face can be hierarchically decomposed according to physical scales representing coarse to fine details.3 Igor Guskov and his colleagues separated the face details from the underlying displaced subdivision surfaces17,18 and then analyzed the resulting displacement map to build a statistical model. New face details are generated by a statistical model using texture synthesis.19 Bernd Bickel and his colleagues presented a multiscale approach to synthesize expression wrinkles.4 Thin shell deformation exploits sparse motion capture markers for generating smooth large-scale mesh deformations.
Then, fine details such as wrinkle deformations are added from the high-resolution captures using nonlinear energy optimizations. Wan-Chun Ma and his colleagues presented a real-time capturing system for high-resolution face geometry using the structured light and photometric stereo. They compute displacements between a neutral mesh deformed by motion capture markers and the high-resolution face geometry in training. These displacements are reconstructed by polynomial displacement maps and used to synthesize facial performances with dynamic wrinkles and fine-scale facial details. Haoda Huang and his colleagues leveraged 3D motion capture and scanning techniques to build a high-resolution face for animation. To build fine-scale details, a sparse set of fine scale expressions is fit into the motion capture sequences using their registration method. Amit Bermano and his colleagues enhanced low-resolution facial performances by adding subtle facial features such as fine wrinkles and pores from a high-resolution temporally coherent performance database. Supasorn Suwajanakorn and his colleagues reconstructed a controllable 3D face model of any person from a photo collection, where the face model can be controlled using a video sequence of a different person. Recently, Chen Cao and his colleagues proposed a regression model for generating face wrinkles and developed a real-time system for capturing facial performances with high fidelity. Most of these methods use special capturing setups to achieve multiscale face details. In contrast, our technique starts from existing dense face models that can be sculpted by artists or obtained from conventional capturing systems.

Representing details using displaced surfaces has been an active research topic in geometry modeling. Since its introduction by Robert Cook, the displaced surface has established itself as a standard shape representation scheme for highly detailed geometric models. Venkat Krishnamurthy and Marc Levoy presented a technique to approximate an arbitrary mesh using vector displacement maps over a network of B-spline patches. Although it yields promising results, their method can produce discontinuities across the patches and displacement map boundaries. To maintain the continuity of the displaced surface, Aaron Lee and his colleagues introduced a displaced subdivision scheme that unifies subdivision surfaces and scalar displacements, and Guskov and his colleagues present a surface representation that successively applies scalar displacement maps to a mesh to be subdivided. In another work, geometry details are encoded in Laplacian coordinates, allowing the detail to be transferred to another surface. Mario Botsch and Leif Kobbelt presented a method for improving both robustness and efficiency of multiresolution shape editing by remeshing the base mesh.

A method that leverages existing face models can save much effort. Shape blending is an especially simple, but powerful approach that is prevalent in facial animation for generating arbitrary expressions. That method provides semantic control that is generally preferable in many film and video game production studios. Ma and his colleagues utilized shape blending as an alternative tool for synthesizing a new face model from existing faces. They segment meaningful face regions (such as the eyes, nose, and mouth) from different faces and spatially assemble them into a new face. Their method requires vertex-wise correspondence across each face and its blendshapes.

We present a novel shape blending scheme to synthesize a new face model using weighted blending of spatial details from various face models. Our MFMs are fully corresponded in a common parameter space, where the artist can interactively define semantic correspondences across faces and scales.

**Face Parameterization and Multiscale Approximation**

The first step in creating a multiscale face model from a given 3D face model \( \mathcal{M} \) is to find its parameterization \( \mathcal{P} \) on a unit domain \([0, 1] \times [0, 1] \subset \mathbb{R}^2\). \( \mathcal{P} \) provides a common parameter space for multiscale displacements as well as intuitive correspondences across faces. We use the convex combination approach, which minimizes area distortion and guarantees a one-to-one embedding.

Let \( \mathbf{v}_i \) be the \( i \)th vertex of \( \mathcal{M} \) and \((u_i, v_i)\) be the corresponding parameters in \( \mathcal{P} \). Then, we can find an approximation surface \( S(u, v) : \mathcal{P} \rightarrow \mathbb{R}^3 \) to \( \mathcal{M} \) by minimizing \( \sum_{i=1}^n |\mathbf{v}_i - S(u_i, v_i)|^2 \), where \( n \) is the number of vertices of \( \mathcal{M} \). In this article, we use the multilevel B-spline technique to construct an approximation surface with hierarchical representations. More specifically, we approximate \( \mathcal{M} \) using a uniform cubic B-spline surface \( F_1(u, v) : \mathcal{P} \rightarrow \mathbb{R}^3 \) with \( 4 \times 4 \) control points. The approximation errors \( e_1^i = \mathbf{v}_i - F_1(u_i, v_i) \) are then refitted by a B-spline surface \( F_2(u, v) \) with refined \( 5 \times 5 \) control points while producing the errors \( e_2^i = e_1^i - F_2(u_i, v_i) \). By repeating this process until the maximum of \( e_i^i \) is within a given tolerance, we can construct an approximation surface \( S(u, v) \) at level \( l \) as follows:

\[ S(u, v) = F_{2l}(u, v), \]
Multiscale CDMs

For each vertex \( v_i \) of \( M \), its projection point \( v'_i \) onto \( S(u, v) \) can be computed by finding the solutions \((u, v)\) of the following equation:

\[
S_i(u, v) = \sum_{l=1}^{k} F_l(u, v) = 0, \tag{1}
\]

where \( 1 \leq l \leq k \). Figures 2b through 2g show the results of approximation to the face model in Figure 2a at different levels.

\[
\begin{align*}
< v_i - S_i(u, v), \frac{\partial S_i(u, v)}{\partial u} > &= 0 \\
< v_i - S_i(u, v), \frac{\partial S_i(u, v)}{\partial v} > &= 0, \tag{2}
\end{align*}
\]

where \( < , > \) is an inner product. Let \( M' \) be the face model projected onto \( S_i(u, v) \). If no self-intersections of \( M' \) exist, the original face model \( M \) can be represented as scalar displacement functions from \( S_i(u, v) \). Based on this observation, we can choose the coarsest surface \( S_d(u, v) \) for sampling scalar displacements, where we force \( d \geq 2 \) to introduce a base surface \( S_b(u, v) \) in Equation 3. Note that \( S_d(u, v) \) maximally captures the face details as scalar displacements, compared with other surfaces \( S_i(u, v) \), \((l > d)\).

Figure 2d shows the chosen surface \( S_d(u, v) \) with normal vector fields. The scalar displacements are sampled by computing the intersections of the normals of \( S_d(u, v) \) with the face model, and they are approximated by multilevel B-spline functions \( f_i(u, v) : P \rightarrow \mathbb{R}, (i = 1, 2, \ldots) \). We then define a CDM \( D(u, v) : P \rightarrow \mathbb{R} \) by manually grouping the displacement functions \( f_i(u, v) \) and merging them.

In this article, we tested our method using three-level scalar CDMs \( D_l(u, v), (l = 1, 2, 3) \), but there is no technical limitation to vary the number of levels. The semantic meaning of the decomposition into different scales is defined by artists based on how they group the displacement functions \( f_i(u, v) \). Figure 3 shows the multiscale CDMs thus constructed, where three-level scalar CDMs...
represent semantic scales of the face details. Note that each $D_l(u, v)$ can have a different group of $f_u(u, v)$ depending on each face model.

**Multiscale Face Model**

Now, the face model $M$ can be reconstructed by adding a set of scalar CDMs $D_l(u, v)$, ($l = 1, 2, 3$) to the surface $S_d(u, v)$. Although these scalar CDMs $D_l(u, v)$, ($l = 1, 2, 3$) can represent face details in a hierarchical manner, they have difficulty expressing relatively coarse details correctly. To include such a coarse-level displacement, we slightly modify the reconstruction using a base surface $S_b(u, v)$, ($b = d - 1$) and its vector displacement $D_0(u, v) = S_d(u, v) - S_b(u, v)$. Then a MFM is defined as follows:

$$M = S_b(u, v) + D_0(u, v) + N(u, v) \sum_{l=1}^{3} D_l(u, v),$$  \hspace{1cm} (3)

where $N(u, v)$ is the unit normal of $S_b(u, v)$. Note that $D_0(u, v)$ is a vector CDM, whereas the others $D_l(u, v)$, ($l = 1, 2, 3$) are scalar CDMs. This setup provides more flexibility for fine tuning by artists using the scalar maps. Figure 4 shows a MFM consisting of a base surface and multiscale CDMs.

**Correspondences across Face Models**

To blend CDMs across different levels as well as different faces, correspondences across all the MMFs must be provided. Mesh registration can solve the problem and generally needs guidance from user-specified corresponding features. If two faces have large shape variations, such as human versus nonhuman, more features are needed. However, assigning many vertex features on the 3D face mesh is a time-consuming and somewhat unnatural task for artists.

To provide semantic correspondences across faces with highly different shapes, as in Figure 1c, we instead focus on interactive controls using feature curves. These feature curves provide an intuitive and high-level control over the dense 3D correspondences on the face’s 2D parameter space. For example, feature curves on salient wrinkles can provide the detailed coherence that is required for transferring the fine wrinkles correlated with the salient wrinkles. Although the feature curves can be arbitrarily chosen, the same curves must be drawn on each face. This results in a semantic correspondence across faces, which allows both human and nonhuman faces to be easily manipulated during blending.

**Feature Curves on Parameter Spaces**

As Figure 5 shows, the user can select and mark meaningful face regions by 2D B-spline curves (feature curves) over the parameter space. These curves can include the eyes, nose, mouth, face boundary (for example, only the interior of the boundary will be blended), and noticeable wrinkles on the face. To facilitate this process, we overlay the parameter space with various visualizations including normals, curvature, wireframes, and displacements derived from the corresponding 3D face model. By constructing an efficient data structure such as a quadtree for the parameterization, we can interactively visualize the 3D curves on the 3D face model corresponding to the 2D feature curves in parameter space. This provides intuitive user controls to edit 3D curves in the 2D parameter space.

**Correspondence across Parameter Spaces**

Let $P^A$ and $P^B$ be the parameter spaces (or parameterization) of the face models $M^A$ and $M^B$, respectively. Once we have marked 2D feature curves on each parameter space, the correspondence between $P^A$ and $P^B$ can be established by constructing a function that smoothly deforms $P^A$ and $P^B$.  

Figure 4. Multiscale face model. A linear combination of a base surface (top left) and multiscale CDMs (bottom) reconstructs a 3D face model (right), where a vector CDM is rendered in RGB color, and multiple scalar CDMs are rendered in gray.
using feature curves. We sample points \((u_i, v_i)\) and \((\hat{u}_i, \hat{v}_i)\) on each curve of \(\mathcal{P}^A\) and \(\mathcal{P}^B\), as in Figures 5a and 5b. A smooth deformation function \(\omega^{AB}(u, v) : \mathcal{P}^A \rightarrow \mathcal{P}^B\) can then be computed from the constraints \(\omega^{AB}(u_i, v_i) = (\hat{u}_i, \hat{v}_i), \forall i\), using a scattered data interpolation technique. Consequently, the correspondence between \(\mathcal{P}^A\) and \(\mathcal{P}^B\) can be derived from \(\omega^{AB}(u, v)\).

As we explained earlier, all the CDMs of a MFM are in a common parameter space. If \(D^A(u, v)\) and \(D^B(u, v)\) are the CDMs defined on the parameter spaces \(\mathcal{P}^A\) and \(\mathcal{P}^B\), respectively, we can generate a deformed CDM \(\hat{D}^A(u, v)\) of \(D^A(u, v)\) using the deformation function \(\omega^{AB}(u, v)\). Figure 5c shows a CDM \(D^A(u, v)\) on the parameter space \(\mathcal{P}^A\), and Figure 5d shows the deformed CDM \(\hat{D}^A(u, v)\), where all feature regions are aligned with those of \(\mathcal{P}^B\) (Figure 5b).

**Blending Multiscale Face Details**

Because we have full correspondences across both face models and scales, a user can interactively synthesize a new 3D face model using weighted blending of the MFM s. Artists can choose CDMs from different levels as well as different faces for blending. For example, we choose a base surface \(S_k^A(u, v)\) of a face model \(M^A\). Then, we can transfer the details from different MFM s using their CDMs. Let \(M^x \in X\) be a face model in a set of MFM s. Multiscale CDMs of \(M^x\) are \(D^l_x(u, v)\), where \(l = 0, 1, 2, ..., N_l - 1\) describes their scale level. The deformed CDMs can be denoted by \(\hat{D}^l_x(u, v)\). Then a user can synthesize a new face model \(M^{\text{new}}\) as follows:

\[
M^{\text{new}} = S^A(u, v) + N^A(u, v) \sum_{M^x \in X} \sum_{l=0}^{N_l-1} \alpha^l_x \hat{D}^l_x(u, v),
\]

where \(S^A(u, v) = S^A(u, v) + \sum_{M^x \in X} \alpha^l_x \hat{D}^l_x(u, v)\) and \(N^A(u, v)\) is the unit normal of \(S^A(u, v)\). At the boundaries of \(M^{\text{new}}\), some artifacts might occur due to the distortions of 2D parameterization. To handle this problem, we can multiply \(\hat{D}^l_x(u, v)\) by a weight function \(w(u, v) = \rho(u)\rho(v)\), where \(\rho()\) can be defined as follows (see Figure 6a):

\[
\rho(t) = \begin{cases} 
-16t^3 + 12t^2 & (0 \leq t \leq 0.5) \\
16(t - 0.5)^3 - 12(t - 0.5)^2 + 1 & (0.5 \leq t \leq 1) 
\end{cases}
\]

Note that \(S^A_k(u, v)\) represents the overall shape of the face model \(M^{\text{new}}\), whereas the face details are generated by weighted blending of selected CDMs from various scales and faces.

Theoretically, if \(N(= |X|)\) is the number of MFM s in \(X\), our method provides \(N \times (1 + N_l)\) degrees of freedom for synthesizing a new face model. Any arbitrary face can be created using \(N\) base surfaces and displacement maps that are blended by \(N \times N_l\) dimensional weight vector \(\vec{\alpha} = [\alpha^A_0, \alpha^A_1, ..., \alpha^A_{N_l-1}]\) in the blendshape space, where \(\vec{\alpha}^x = [\alpha^A_0, \alpha^B_0, ..., \alpha^B_{N_l-1}]\).

**Results**

We initially tested our method using a set of 15 different face models that were captured from either a 3D scanner or sculpted by artists. The set consists of 10 human faces and five nonhuman faces including several animals, a monster, and an
alien. The number of vertices of each face model varies from 21,000 to 677,000.

Figure 7 shows the MFMs of the 15 faces. Each MFM consists of a base surface and four CDM levels. For notational convenience, we named each MFM \( M^A, M^B, \ldots, M^O \) in sequential order. We then tested our methods using an Intel Core2 Duo 2.80-GHz CPU with a 4-Gbyte main memory and an Nvidia GeForce 9600M GT video card. We can interactively blend MFMs without additional hardware acceleration.

**Blending Multiscale Human Faces**

A new face model is synthesized by weighted blending of the MFMs. Because we used 15 faces consisting of a base surface and four-scale CDMs, we have a \( 15 + 15 \times 4 \) dimensional blendshape space; the total number of blendshapes can be limited by the artist. Any arbitrary face model can be created by 15 base surfaces and CDMs blended by a weight vector \( \mathbf{\alpha} \). Our method does not have any restriction in terms of the number of faces and scale levels of the CDMs in the associated blendshape space.

Figure 8 shows examples of synthesized faces using multiple human face models. The face in Figure 8a is synthesized using a base surface \( S^b_b(u,v) \) of face \( M^b \) and the CDMs \( D^b_H(u,v) \) of face \( M^H \) using weight vectors \( \mathbf{\alpha}^b = (0.5, 0, 0, 0) \) and \( \mathbf{\alpha}^H = (0.5, 1, 1, 1) \). Figures 8b and 8c show results of using two faces with different blending weights. The faces are constructed using a base surface \( S^b_b(u,v) \) of face \( M^a \) and the CDMs \( D^d_D(u,v) \) of face \( M^D \) using the blending weight vectors \( \mathbf{\alpha}^a = (0.5, 0, 0, 0), \mathbf{\alpha}^d = (0.5, 1, 1, 1) \) for Figure 8b and \( \mathbf{\alpha}^a = (1, 1, 0, 0), \mathbf{\alpha}^d = (0, 0, 1.5, 1.5) \) for Figure 8c. As with conventional blendshapes for facial animation, the sum of the blending weight of each level does not need to be normalized to 1.0, allowing more artist control.

Figures 8d through 8g show examples of synthesized face models using weighted blending of three MFMs. In Figure 8d, we choose the base surface \( S^b_b(u,v) \) of \( M^b \) and then added the CDMs \( D^H_H(u,v) \) of face \( M^H \) and \( D^D_D(u,v) \) of \( M^D \) to \( S^b_b(u,v) \) using blending weights \( \mathbf{\alpha}^b = (0.2, 0, 0, 0), \mathbf{\alpha}^H = (0.8, 1, 0, 0), \) and \( \mathbf{\alpha}^D = (0, 0, 1, 1) \), respectively. Consequently, the resulting face model obtains the face proportions from \( M^b \), with regions such as the nose, mouth, and salient wrinkles taken from \( M^H \), and fine details such as eyelids, fine wrinkles, and pores from \( M^D \). The other results in Figures 8e through 8g were created using a similar method.

Table 1 lists times for synthesizing new face models in Figure 8. As these results show, our method supports interactive blending of different MFMs.

**Detail Transfer from/to Nonhuman Faces**

Our method is not restricted to human faces. Transferring details between sculpted nonhuman creatures and a human actor’s face is an important task in visual effects. Figures 9a through 9c show our results in transferring face details between a nonhuman face and a human face. In
Figure 9a, the details of the monster face $M^t$ were transferred to the human face $M^o$. The artist-sculpted details were nicely transferred to the human face without artifacts. Figure 9b shows a result of transferring details from an alien face $M^a$ to a human face $M^o$. Because of the different shapes of the nonhuman faces, our approach only transfers inside of the face boundary defined by feature curves; the region outside the boundary is blended using a Gaussian blur.

Transferring details from a human face to a nonhuman is particularly important in the case of a werewolf. The challenge here is to preserve the shape of the nonhuman character while presenting some of the human face's identity. We can drive the solution by controlling the detail blending between the human and nonhuman faces. In Figure 9c, we choose a base surface from an armadillo face $M^o$. Then, the details of the human faces $M^o$ were transferred to the armadillo face using the CDMs $D^o(u,v)$ blended by weight vectors $\alpha^o = (0.5, 0, 0, 0)$ and $\alpha^H = (0.5, 1, 1, 1)$, whereas (b, c) the following two examples show results of using two faces with different blending weights. (d–g) The next set shows examples of synthesized face models using weighted blending of three MFM.

### Table 1. Synthesizing times for the face models in Figure 8.*

<table>
<thead>
<tr>
<th>Figure</th>
<th>Base surface construction</th>
<th>Evaluation warping functions</th>
<th>Adding vector displacements</th>
<th>Adding scalar displacements</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>8d</td>
<td>68</td>
<td>72</td>
<td>48</td>
<td>21</td>
<td>209</td>
</tr>
<tr>
<td>8e</td>
<td>68</td>
<td>93</td>
<td>57</td>
<td>32</td>
<td>250</td>
</tr>
<tr>
<td>8f</td>
<td>82</td>
<td>93</td>
<td>68</td>
<td>31</td>
<td>274</td>
</tr>
<tr>
<td>8g</td>
<td>49</td>
<td>55</td>
<td>52</td>
<td>22</td>
<td>178</td>
</tr>
<tr>
<td>Average</td>
<td>66.75</td>
<td>78.25</td>
<td>56.25</td>
<td>26.50</td>
<td>227.75</td>
</tr>
</tbody>
</table>

*All results are in milliseconds.

Figure 8. Blending multiscale human faces. (a) The first face was synthesized using a base surface $S^B(u,v)$ of face $M^H$ and the CDMs $D^H(u,v)$ of face $M^H$ using weight vectors $\alpha^B = (0.5, 0, 0, 0)$ and $\alpha^H = (0.5, 1, 1, 1)$, whereas (b, c) the following two examples show results of using two faces with different blending weights. (d–g) The next set shows examples of synthesized face models using weighted blending of three MFM.

**Detail Transfer to Ambiguous Face Models**

The multiscale face blending together with feature curve controls in the parameter space provides a
practical solution for transferring face details to an abstract face. For example, face regions such as the eyes, nose, and mouth on the Stanford bunny model are somewhat ambiguous, as Figure 10a shows. Automatically computing correspondences between such a face and other faces is challenging. Interactive user manipulation provides a practical solution in such cases.

Users can select a specific region $R$ to transfer details to (see Figure 10a). Then, an appropriate parameterization (parametric space) $P_R$ of $R$ is computed, as we described earlier. Interactive manipulation of 2D feature curves on the 2D parameter space of $P_R$ assigns the semantic meaning of the ambiguous face (see Figure 9c). Then, the parameter space provides the desired correspondences with the other. Figure 10b shows the Stanford bunny model with transferred details from the face models $M_C$ and $M_D$.

Furthermore, interactive control of the 2D feature curves provides an intuitive user interface on the 2D parameter space (Figure 10d) to manipulate face details on the ambiguous face. As a result, we produced a facial animation of the Stanford bunny using a detail transfer (see Figure 10e). Note that this was not possible using the original model because it did not have explicit eyes or a mouth to manipulate.

Because our multiscale face models have full semantic correspondence across both identities and scales, we can reliably produce quality transitions simply by using linear interpolation of blending weights. See the accompanying video at youtu.be/zrUjvJj07uY for examples.

**Feedback from Artists**

Lastly, we conducted a subjective test with 10 studio artists. We asked the artists to use our method to synthesize a new face using the different faces in multiple scales. After choosing a base surface, they added details from two or three different models while adjusting blending weights interactively. We gave the artists 10 minutes to perform these tasks.

After the artists finished editing, we asked them two questions:

- **Q1:** Could you synthesize a new face using the proposed method and if it was easy and intuitive to use, the artists answered affirmatively, with a mean score of around 4 (yes).
Question 1: Given this method, could you synthesize a new face?

Question 2: Is the new modeling method easy and intuitive to use to create a new face?

They answered with a score between 1 and 5, where 1 is definitely no, 2 is no, 3 is uncertain, 4 is yes, and 5 is definitely yes. The mean score of the qualitative evaluation by these professional artists was around 4 (yes) for both questions (see Figure 11).

There are several issues to be considered in future work. An obvious extension would be to provide local region blending. We can simply simulate this using a local face mask on the parameter space. In our current pipeline, correspondences across MFMs are provided using feature curves defined in the parameter space. To interactively handle ambiguous and non-human faces, we focus on artist controls rather than formulating an optimization problem that might run slowly and fail occasionally. However, our method does not prohibit adding such a step. Trade-offs between interactive semantic controls and automated optimization would be a good discussion point. A solution to satisfy both targets would be ideal and is left to future study.

Acknowledgments

This work was supported by the ICT (Information and Communications Technologies) R&D program of MSIP/IITP (R0190-16-1113, Developing Digital Dentistry System for Implant Surgical Guide and Prosthetic Design).

References

20. H. Huang et al., “Leveraging Motion Capture and


Seung-Hyun Yoon is an associate professor in the Department of Multimedia Engineering at Dongguk University. His research interests include computer graphics and geometric modeling. Yoon has a PhD in computer science and engineering from Seoul National University. Contact him at shyun@dongguk.edu.

John Lewis is an adjunct associate professor at the Victoria University of Wellington. His research interests include machine learning and computer vision applied to computer graphics. Lewis has a PhD in computer science from Victoria University. He is a member of IEEE and ACM. Contact him at zilla@computer.org.

Taehyun Rhee (corresponding author) is an associated professor and deputy director of the Computational Media Innovation Center at the Victoria University of Wellington and a founder and director of the Victoria Computer Graphics Lab. His research interests include perceptually based lighting and rendering, mixed-reality rendering, immersive visualization on HMDs, realistic human body modeling and animation, and human digital content interaction. Rhee has a PhD in computer science from the University of Southern California. Contact him at taehyu.rhee@ecs.vuw.ac.nz.