Fabricating 3D Figurines with Personalized Faces

J. Rafael Tena, Moshe Mahler, Thabo Beeler, Max Grosse, Hengchin Yeh, and Iain Matthews ▪ *Disney Research*

A long-standing goal in business is to provide personally tailored goods and services. Many companies face demands for personalized products.[1] Although consumer-tailored services have become ubiquitous, the elevated costs of producing one-of-a-kind prototypes have comparatively hindered the delivery of customized fabricated goods. Advances in 3D printing could enable the fabrication of highly customized goods for a fraction of the traditional cost, opening new avenues for research and commercial enterprise. One opportunity is to use the consumer's face to personalize artifacts. In particular, companies are producing customized figurines, such as bobble heads and action figures, with faces that intend to capture the customer's likeness.

A new system seamlessly integrates 3D facial data with a predefined figurine body into a unique, continuous object that's fabricated as a single piece. Human intervention is limited to a few simple tasks to maintain the high throughput and consistent quality required for commercial application.

We present a semiautomated system for fabricating figurines with faces that look like the customer's. Our system is robust and has been deployed commercially. During its deployment, young girls could have their facial likeness captured in a figurine of a Disney princess of their choice. General customers could have their facial likeness frozen in "carbonite," like in the classic scene in *Star Wars Episode V: The Empire Strikes Back*. Figure 1 shows some example figurines.

Other products, such as those from Clone Factory Inc. (http://clonefactory.co.jp) and HeadBobble.com (www.headbobble.com), consist of fabricated custom heads attached to prefabricated plastic bodies or action figures. In contrast, our system seamlessly integrates scanned 3D facial data with a predefined figurine body into a unique, continuous object that's fabricated as a single piece by 3D printing. Human intervention is limited to a few simple tasks to maintain the high throughput and consistent quality required for a commercial application. The system's core components—state-of-the-art 3D capture, modeling, and printing—provide the flexibility to fabricate figurines whose complexity is limited only by the designer's creativity.

This is a complete system that fabricates high-quality products at a commercial volume. It's the only system to date that can mass-produce fully integrated, seamless, single-piece figurines.

Figurine Modeling

The system requires a human-like computer-generated (CG) 3D figurine M whose face will capture the customer's likeness. Offline, a professional 3D modeler creates a CG model—for example, a figurine of Belle from Disney's *Beauty and the Beast* (see Figure 2). The figurine's face has generic human features, and its body matches a young girl's anatomy. The CG model is expressed as a triangulated mesh. That is, its surface is approximated by a set of triangle primitives, each of which is described by its three vertices' Cartesian coordinates.

The modeler also creates a texture map T, an image describing each triangle primitive's color. He or she does this by mapping each triangle to the image plane and filling the corresponding pixels with the desired color. Figure 2 shows T for the Belle figurine.

After completing M, the modeler extracts from it a face template M_G, a pivot template M_P, and a

 Published by the IEEE Computer Society

Figure 1. Figurines fabricated by our system. The system seamlessly integrates 3D facial data with a predefined figurine body into a unique, continuous object that's fabricated as a single piece by 3D printing.

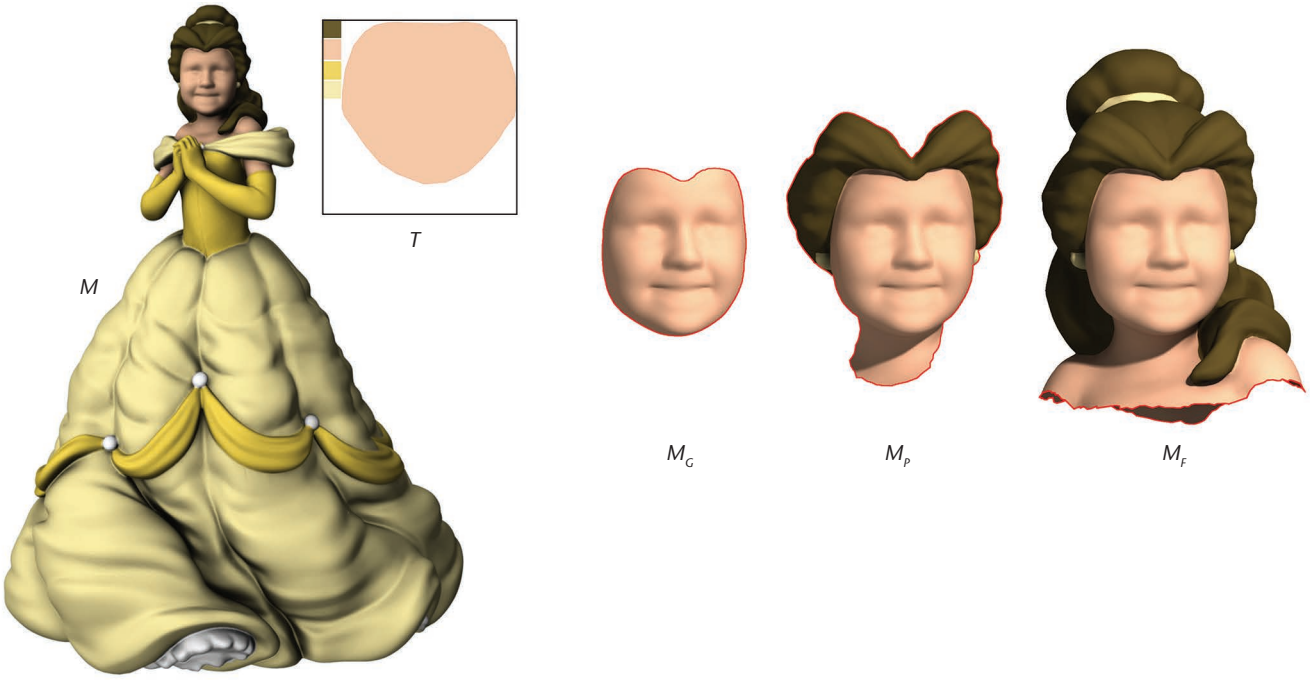

Figure 2. A professional 3D modeler created a digital figurine (*M*) of Belle from Disney's *Beauty and the Beast* and its corresponding texture map (*T*). The figurine has generic facial features and a young girl's anatomy. The modeler also extracted a face template (M_G), pivot template (M_P), and fixed template (M_F). The system uses the boundary vertices (in red) to consistently integrate the customer's facial 3D scan and the digital figurine.

fixed template M_F (see Figure 2). Each template is a subset of M. The system uses the templates to guide different processes and provide an additional layer of quality control, as we show later.

Table 1 lists symbols we use throughout the rest of the article for the figurine, its templates, and other 3D CG assets.

Creating and Fabricating Figurines

The system comprises five steps:

1. Acquire the 3D geometry and color of the customer's face.
2. Register M_G to the customer's 3D data M_T.
3. Attach M_G to the body template M_B.
4. Color-match the registered texture map T_R to be consistent with the style of T.
5. 3D-print the customized figurine M_{RF}.

Figure 3 diagrams the system.

3D Facial Capture

To capture the customer's facial geometry and color, we use Thabo Beeler and his colleagues' single-shot 3D scanner, which employs multiple cameras and passive stereo.[2] It captures the 3D

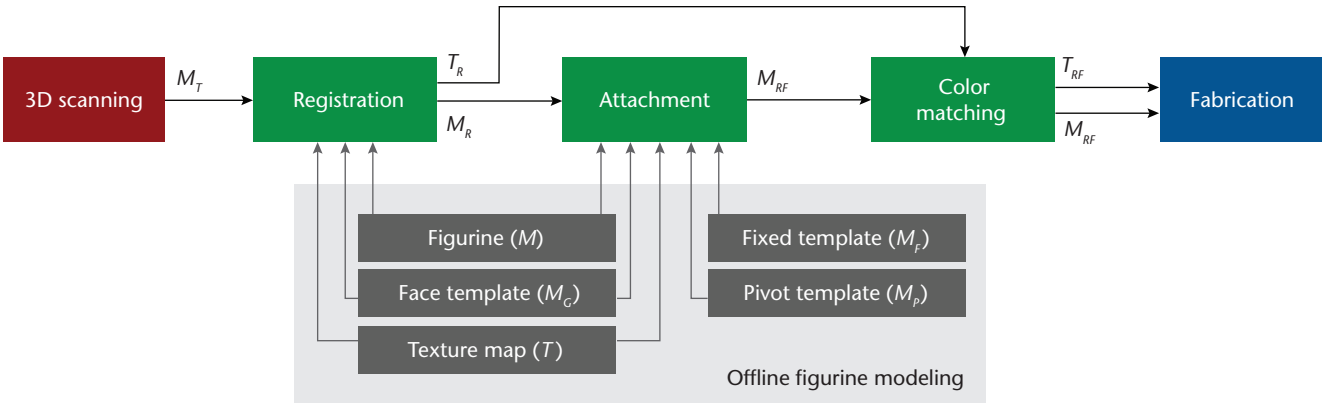

Figure 3. A system for fabricating figurines with personalized faces. The colored boxes show the process steps; gray boxes are assets generated offline. Arrows represent flows of assets through the pipeline. For an explanation of the symbols, see Table 1.

Table 1. Symbols used for the assets that flow through our system's pipeline.

Asset	Symbol
Customer's facial 3D scan	M_T
CG figurine	M
CG figurine's texture map	T
CG figurine's face template	M_G
CG figurine's pivot template	M_P
CG figurine's fixed template	M_F
CG figurine's body template	M_B
Registered face template	M_R
Registered texture map	T_R
Customized CG figurine	M_{RF}
Customized CG figurine's texture map	T_{RF}

Figure 4. The top image is a customer's 3D facial scan (M_T). We used Thabo Beeler and his colleagues' 3D scanner,[2] which used the images in the bottom row to reconstruct the face's 3D geometry and color.

geometry and color of the customer's face in a single shot under standard light sources, with submillimeter accuracy. It outputs the geometry as a triangulated mesh with per-vertex colors specified as RGB triplets. A typical scan contains over half a million vertices. Unlike laser-based 3D scanners, computer-vision-based scanners like this one don't require the subject to remain still for a prolonged time. This characteristic makes them suitable for capturing the faces of customers of any age and allows for fast, high-throughput acquisition.

Figure 4 shows 3D data captured by our system. The scanner simultaneously captures eight photographs, which it uses to reconstruct the face's 3D geometry. Our system isn't constrained to a particular 3D acquisition device; it can employ any method or hardware that accurately captures 3D geometry and color.

Registration

Registration establishes point-to-point correspondences between two 3D objects. We adapted J. Rafael Tena and his colleagues' nonrigid template-based 3D facial registration.[3] Nonrigid template-based methods typically optimize an energy function that deforms the template into the target object's shape. In our case, we want to deform M_G into the shape of M_T. To guarantee high registration accuracy even for unpredictable facial expressions, a human operator annotates a few key feature points, such as the eye and mouth corners, on M_G and M_T.

Tena and his colleagues' method comprises three steps: *global mapping*, *local matching*, and *energy minimization*. Global mapping uses thin-plate-spline (TPS) interpolation.[4] TPS interpolation warps M_G to align its k annotated feature points to the corresponding set of annotated feature points on M_T. For example, assume that $\mathbf{p}_i = (x_{pi}, y_{pi}, z_{pi})$ is the ith feature point on M_G and that $\mathbf{q}_i = (x_{qi}, y_{qi}, z_{qi})$ is the corresponding feature point on M_T. If the distances between the feature points on M_G

are $d_{ij} = |\mathbf{p}_i - \mathbf{p}_j|$, the interpolating function that warps $\mathbf{p}_i$ to $\mathbf{q}_i$ is

$$F(x,y,z) = \left[f_x(x,y,z), f_y(x,y,z), f_z(x,y,z)\right],$$
$$f_{k=x,y,z}(x,y,z) = a_1^k + a_x^k + a_y^k y + a_z^k z + \sum_{i=1}^{n} w_i^k B\left(\left|\mathbf{p}_i - (x,y,x)\right|\right),$$

where $B(d)$ is a biharmonic function ($B(d) = d$ in our implementation), and $\left(\mathbf{w}^k, a_1^k + a_x^k + a_y^k + a_z^k\right)^T = f\left(B(d_{i,j})\mathbf{p}, \mathbf{q}\right)$ are the TPS interpolant's parameters. Fred Bookstein described these parameters' computation in detail.[4]

Local matching finds for each vertex $\mathbf{v}_i$ in M_G a corresponding point on the surface of M_T. Tena and his colleagues defined this point as the closest vertex $\tilde{\mathbf{v}}_i$.[3] Instead, we define the corresponding point $\tilde{\mathbf{s}}_i$ as the intersection of a ray, projected from $\mathbf{v}_i$ along the surface normal, and the surface of M_T. Because M_T is a triangulated mesh, we can use *barycentric coordinates*[5] to describe any point on its surface. If the Euclidean distance between $\mathbf{v}_i$ and $\tilde{\mathbf{s}}_i$ is greater than 1/10 of the length of M_T or if their surface normals differ by more than 90 degrees, we flag the correspondence of $\mathbf{v}_i$ to $\tilde{\mathbf{s}}_i$ as unreliable.

We used experimentation to derive our choices of thresholds for the Euclidean distance and surface normal to determine correspondences' reliability. The surface normal variation across individuals can be large in the nose region. A threshold of 90 degrees prevents matching surfaces that have perpendicular orientations but allows for a wide range of nose variation. We observed in our testing data that 1/10 of the length of M_T was a conservative estimate to separate genuine facial data from possible reconstruction errors.

Finally, energy minimization optimizes an objective function with two terms. The data term E_{ext} penalizes the Euclidean distance between the correspondences determined during local matching. The smoothness term E_{int} penalizes deformations of M_G.

This gives us

$$E_t = E_{ext} + \lambda E_{int}, \quad (1)$$

$$E_{ext} = \sum_{i=1}^{n} w_i \left\|\tilde{\mathbf{s}}_i - \mathbf{v}_i\right\|_2^2,$$

$$E_{int} = \sum_{i=1}^{n}\sum_{j=1}^{m} \left\|\left(\mathbf{v}_i - \mathbf{v}_j\right) - \left(\bar{\mathbf{v}}_i - \bar{\mathbf{v}}_j\right)\right\|_2^2,$$

where λ is a constant that weights each term's contribution, n is the number of vertices in M_G, w_i is a weight set to 0 for correspondences flagged as unreliable and 1 otherwise, m is the number of vertices $\mathbf{v}_j$ connected by an edge to $\mathbf{v}_i$, and $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{v}}_j$ are those vertices' original positions in M_G. In other words, E_{ext} pushes the vertices of M_G toward their corresponding points on the surface of M_T, whereas E_{int} constrains neighboring vertices in M_G to maintain their positions relative to each other. Accordingly, λ controls the rigidity of M_G.

```
Annotate feature points on M_T
Global mapping
for i = 1 to 2 do
  Local matching
  Energy minimization
  Filter M_G
  Reset v̄ to the current v values
end for
Transfer texture from M_T to M_R
```

Figure 5. The registration algorithm. We register the face template (M_G) to the customer's 3D data (M_T). M_R is the registered face template; $\bar{\mathbf{v}}$ is the original position of vertex $\mathbf{v}$ before energy minimization.

Like Tena and his colleagues, we minimize the objective function using the conjugate-gradient method.[3] Specifically, we optimize Equation 1, apply Gabriel Taubin's nonshrinking filter[6] to further regularize M_G, set the values of $\bar{\mathbf{v}}$ to the current positions of $\mathbf{v}$, and repeat local matching, energy minimization, and filtering. Applying the nonshrinking filter to M_G lets us use lower values of λ—that is, M_G is less rigid—while maintaining its smoothness. The registration process produces the registered template M_R, which has the mesh topology of M_G but the 3D shape of M_T. Figure 5 shows the registration algorithm.

After registration, we transfer the color of M_T to M_R by texture mapping. Each vertex of M_G is parameterized by a (u, v) pair that maps its 3D coordinates to a point in the 2D image. Because M_G and M_R have the same topology, they both map to the same (u, v) pairs.

Accordingly, for each triangle of M_R in (u, v) space, we find all the pixels $\mathbf{t}_i$ of T that are in the triangle. Using barycentric coordinates, we find for each $\mathbf{t}_i$ its corresponding 3D coordinate $\mathbf{s}_i$ on the surface of M_R. We project a ray from $\mathbf{s}_i$ onto the surface of M_T to find the corresponding point $\tilde{\mathbf{s}}_i$ and its barycentric coordinates (M_T is also a triangulated mesh). We use the vertex colors of the triangle on which $\tilde{\mathbf{s}}_i$ lies, along with the point's barycentric coordinates, to find the RGB color value assigned to $\mathbf{t}_i$.

Figure 6 shows an example of registration and texture extraction. The set of feature points used for global mapping are in blue.

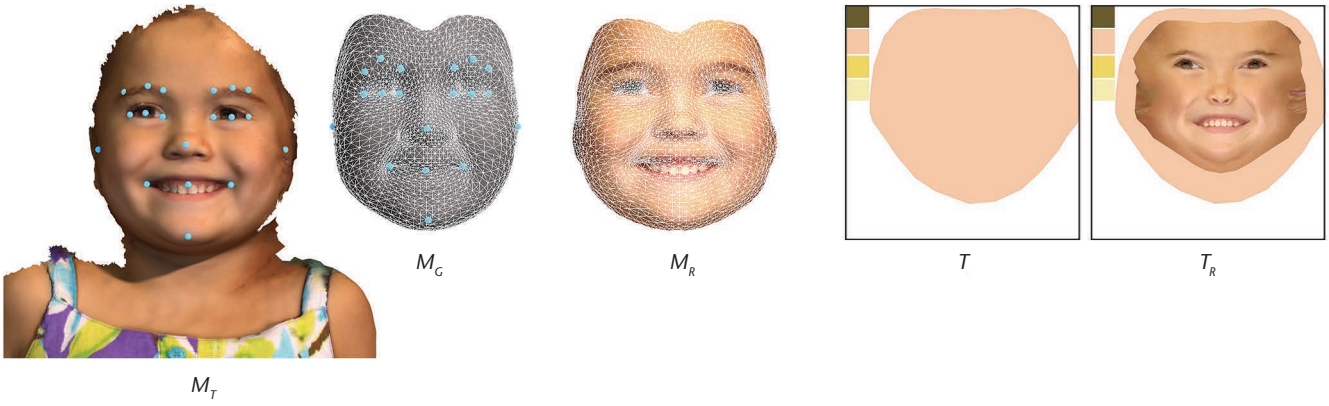

Figure 6. The system registers the face template (M_G) to the customer's face scan (M_T) and creates the registered template (M_R). The system also extracts the color information of the customer's 3D scan and adds it to the figurine's texture map (T) to generate the registered texture map (T_R). The feature points (in blue) initialize the registration algorithm.

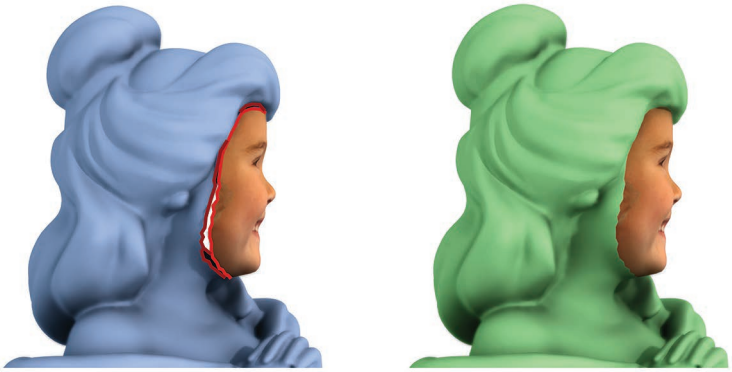

Figure 7. The system uses thin-plate-spline interpolation to warp the body template (M_B) to fit it to the registered template (M_R). The body template before the warp is in light blue. The fully assembled figurine (M_{RF}) is in green. The boundary vertices in this figure (in red) and the boundary vertices of the fixed template M_F and pivot template M_P (in red in Figure 2) serve as control points for the warping interpolant.

We experimented with M_G ranging from approximately 3,000 to 9,000 vertices. For our figurines' size and the 3D printers' resolution, the lower end of the range was sufficient. The system automatically removes the triangle primitives corresponding to M_G from M to create the body template M_B. Mathematically, $M_G \cup M_B = M$ and $M_G \cap M_B \neq \varnothing$, because M_G and M_B share vertices at their boundaries (see Figure 7).

A human operator manually annotates a set of 19 feature points on M_T, which have also been pre-annotated offline on M_G (see Figure 6). This annotation is the only human intervention required for registration.

The system applies *ordinary Procrustes analysis*[7] on the set of feature points to find the transformation that scales and aligns M_T to M_G. After alignment, the registration algorithm deforms M_G into the shape of M_T to create M_R and the registered texture map T_R (see Figure 6). We set λ to 0.25 to emphasize E_{ext} and obtain greater deformation.

Finally, we randomly select a set of 1,000 vertices on M_G and M_R; the system uses Procrustes analysis to align the latter to the former. This step ensures that the scale and orientation of M_R correctly match M_B. We chose 1,000 vertices as an empirical compromise that provides enough accuracy while reducing the computation time.

Our system could use any of the several available 3D-registration methods. The insight is to use 3D nonrigid registration to transfer the information in M_T to M. Tena and his colleagues validated their method on a set of 912 3D facial scans.[3] Despite having a simple objective optimization function, the method is effective for our purposes.

Its major drawback is the need for a subset of correspondences, which are provided manually, for initialization. This feature is common to registration algorithms and can be addressed with automatic feature detection. However, these algorithms rarely attain 100 percent detection. Furthermore, their accuracy degrades when the faces have extreme expressions, are of varying ethnic backgrounds, or cover a wide range of ages.

Although this performance is acceptable for many applications, we require 100 percent detection with millimeter accuracy regardless of ethnicity and facial expression. Errors in feature point detection affect the estimation of the transformation that scales and aligns M_T to M_G. This results in defective figurines with incorrect head sizes and facial orientation.

Assigning feature detection to a human operator is currently the safest approach to ensure consistent quality. Given that a trained operator can perform the task in less than a minute, the time savings resulting from the operator correcting automated feature detection, instead of performing the task unaided, is negligible compared to the fabrication time.

Attachment

M_R and M_B are complementary and have corresponding vertices on their boundaries. However, we can't directly join them because their shapes don't match (see Figure 7).

Several methods exist for mesh editing and deformation.[8] Gradient-domain methods are some of the more favored ones. For example, Olga Sorkine and her colleagues[9] and Yizhou Yu and his colleagues[10] used these methods to merge two meshes. Because the boundaries of M_R and M_B have corresponding vertices, there's a one-to-one mapping to create a seam. This allows for a simpler, less computationally expensive solution.

To close the gap between M_R and M_B, we again use TSP interpolation to warp M_B onto M_R. This warping ensures that M_R isn't deformed, which preserves the customer's facial identity. To further restrict the interpolant, we include M_P and M_F. The boundary vertices of M_P and M_F (in red in Figure 2) and M_R and M_B (in red in Figure 7) all serve as control points for the TSP interpolant.

Let $\mathbf{p}_R$, $\mathbf{p}_B$, $\mathbf{p}_P$, and $\mathbf{p}_F$ be the sets of vertices at the boundaries of M_R, M_B, M_P, and M_F, respectively. Then, we find the interpolant $F(x, y, z)$ that maps $(\mathbf{p}_B, \mathbf{p}_P, \mathbf{p}_F)^T$ to $(\mathbf{p}_R, \mathbf{p}_P, \mathbf{p}_F)^T$. This interpolant warps M_B while minimizing the deformation around $\mathbf{p}_P$ and $\mathbf{p}_F$.

In practical terms, M_P and M_F let the modeler control which section of the figurine to warp to accommodate the customer's facial geometry. For our Belle figurine, M_P and M_F in Figure 2 ensure that all the vertices below the figurine's shoulder line aren't warped and that the overall shape of Belle's characteristic hairstyle is preserved. Figure 7 shows M_B before and after warping. Once the warping is done, the system fuses the boundary vertices of M_R to those of M_B to create a single manifold mesh, which is the completed customized figurine M_{RF}.

Finally, the system uses Jörg Vollmer and his colleagues' Laplacian filter[11] on the fused vertices to smooth the seam. The filtering proceeds iteratively, which prevents denting. On the first pass, the system applies the filter to the seam vertices and their second-order neighbors—that is, vertices that are separated from the seam by up to two edges. The second pass applies filtering to the seam vertices and their first-order neighbors, and the third and final pass applies filtering only to the seam vertices.

The attachment is fully automatic. We usually define M_P to include half of the head and neck in the coronal plane. This ensures that the warping function maintains the head's shape. We define M_F to include everything above the figurine's shoulders to preserve the body's overall shape. To avoid smoothing sharp geometric details intended by the modeler, before filtering the seam, the system extracts from T, which contains the original figurine colors before registration, the colors of the vertices selected for filtering and their first-order neighbors. A vertex with a neighbor that has a different color is at a detail boundary and therefore isn't filtered.

Because the figurines' faces are small, high-frequency details on the texture can exceed the fabrication resolution and combine into spots of unexpected color.

Color Matching

Color matching depends on the figurine's style. For the princess figurines, we must match their hair color and skin tone to those of the customer. Conversely, for the carbonite figurines, we must shift the customer's skin tone to gray. These examples represent the color-matching paradigms. The first modifies some of the figurine's properties to match the customer; the second adjusts some customer features to match the figurine. Accordingly, color matching aims to adjust T to give it an aesthetically consistent appearance.

There are issues inherent to capture and fabrication. For example, the lighting used during face capture produces shading that gets baked into T. This shading is undesirable because we want the fabricated figurine to display the effects of natural ambient light. Also, because the figurines' faces are small, high-frequency details on the texture can exceed the fabrication resolution and combine into spots of unexpected color. In addition, the current 3D-printer color gamut doesn't span all human skin tones and hair colors. So, if T uses out-of-gamut skin tones, the system will fabricate unappealing figurines. Our color matching takes these issues into account.

To solve the problem of image blending and skin tone selection, we exploit the structure of T, which is known a priori and is consistent for all figurines of the same style. During development, we experimented with an automated color-matching pipeline (see Figure 8). In that pipeline, a binary segmentation mask T_f specified the region of T corresponding to the face. A blending mask T_b, which an artist designed, defined the preferred blending

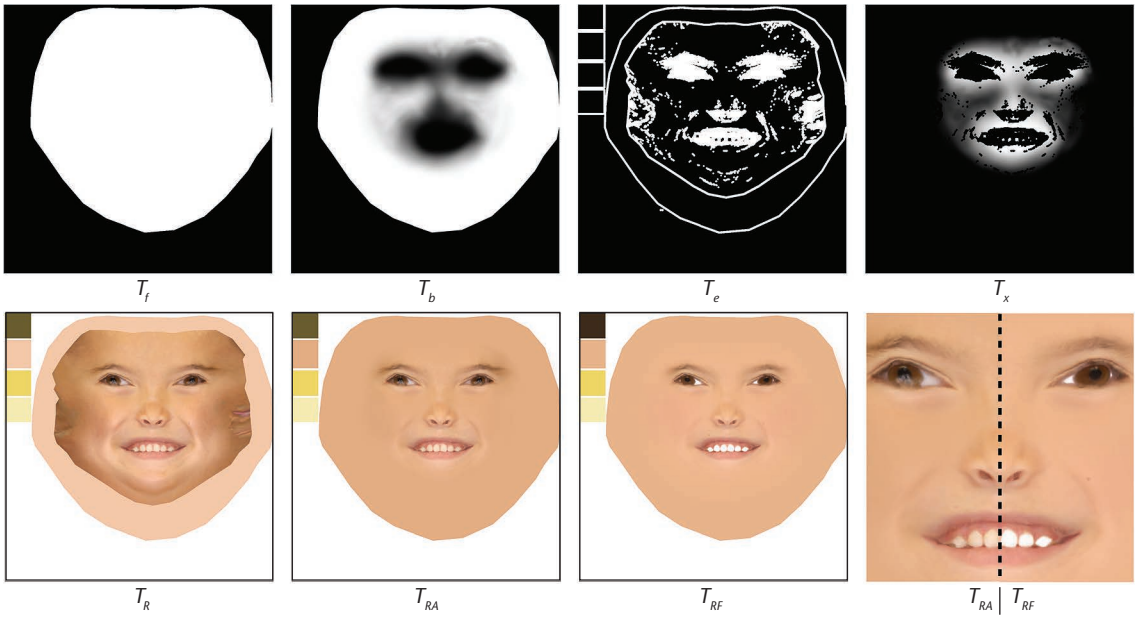

Figure 8. Automated color matching. The system combined a segmentation mask T_f and a blending mask T_b, which were both precomputed, with an edge mask T_e to create a skin tone extraction mask T_x. After extracting the skin tone, we used T_b to blend it into the registered texture map T_R, which resulted in the color-matched texture T_{RA}. For comparison, the figure shows the color-matched texture T_{RF}, which an artist produced. Besides skin tone blending, the artist whitened the sclera and teeth and painted the iris and pupil.

weights. The system automatically computed a binary mask T_e, applying a Sobel edge detector to T followed by dilation, which coarsely masked the facial areas that weren't skin.

The system then combined the three masks to create the skin tone extraction mask T_x. For each pixel i,

$$T_x(i) = (1 - T_b(i)) \cdot T_f(i) \cdot (1 - T_e(i)).$$

T_x emphasized pixels that have low blending weights, which contained reliable skin tone values. We extracted the skin tone value $\mathbf{S}_{rgb} = (S_r, S_g, S_b)$ by applying T_x to T_R. For the red channel, we had

$$S_r = \frac{\sum_i T_{Rr}(i) \cdot T_x(i)}{\sum_i T_x(i)},$$

where T_{Rr} is the red channel of T_R. We obtained S_b and S_g in the same way. Finally, we obtained the skin blended texture map using T_b and $\mathbf{S}_{rgb}$. For each pixel i in the red channel,

$$T_{RAr}(i) = T_x(i) \cdot S_r + (1 - T_x(i)) \cdot T_{Rr}(i),$$

where T_{RAr} is the red channel of the automatically color-matched texture T_{RA}. We obtained T_{RAb} and T_{RAg} in the same way.

Although this pipeline's results were sufficient to create a figurine, it didn't address the aspects of color matching that require higher-level reasoning to improve aesthetics and the likeness to the customer. For instance, whitening the sclera and teeth accentuates the figurine's eyes and smile. However, this requires reliable recognition and segmentation of the eyes and teeth; errors can adversely affect the shape of the eyes and mouth. Also, features such as birthmarks and freckles might be important to the customer's likeness. However, automatically separating these features from unwanted shading or selecting them for contrast boosting is difficult. Moreover, temporary features such as a bruise or sunburn can inappropriately bias the automated skin tone selection.

For the same reasons, Patrick Pérez and his colleagues' Poisson blending[12] would be insufficient. Our pipeline could use Poisson blending to eliminate T_b. However, this wouldn't address the other necessary aesthetic operations, such as whitening and selective contrast boosting, that require higher-level reasoning.

Having an artist in the loop to perform color matching provides better quality and consistency while only slightly affecting throughput. Additionally, it makes the system more adaptable for producing new figurine styles because there's no need to design a corresponding automated color-matching algorithm. The artist can make aesthetic decisions per customer, such as boosting the contrast of features such as eyebrows and birthmarks or even exaggerating them to ensure they remain visible on the fabricated figurine.

During modeling, a lead artist specifies the color-matching protocol for a given figurine style.

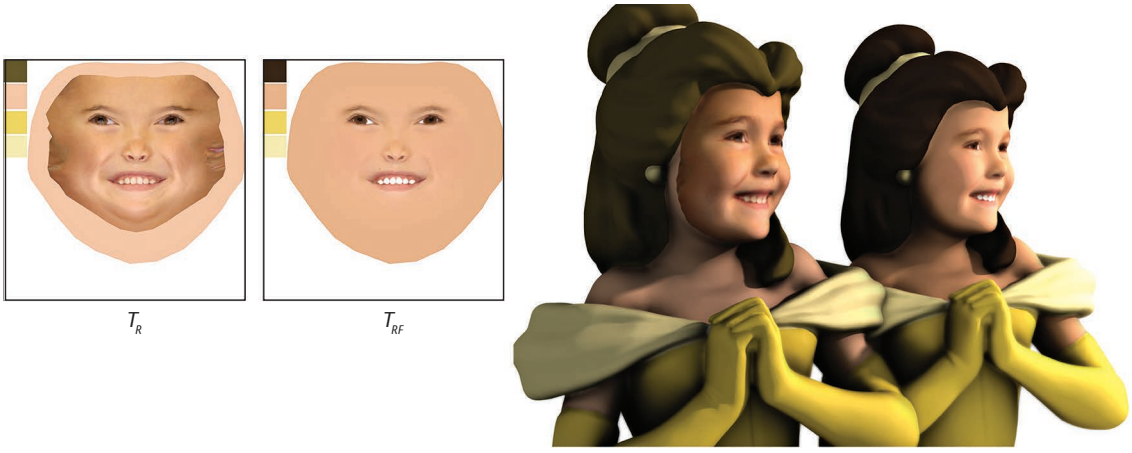

Figure 9. Color matching. An artist edits the registered texture map (T_R) to create the customized figurine's texture map (T_{RF}). The figure shows the customized figurine (M_{RF}) with the two maps. The artist edits the skin tone and the system adapts the hair color so that they match the customer.

For the princess figurines (see Figure 9), this protocol consists of

- whitening the sclera and teeth to increase contrast,
- selecting a skin tone that best approximates the customer's, and
- blending that skin tone into *T*.

During face capturing, the customer selects the figurine's hair color from a preapproved set of in-gamut tones; the system automatically applies it to *T*. To avoid out-of-gamut skin tones, the artist has a chart of preapproved colors. He or she can boost contrast and details to improve the figurine's likeness to the customer.

For the carbonite figurines, the protocol consists of

- adjusting the RGB balance of *T* to gray and
- blending in a gray tone.

In both cases, the artist performs color matching in Adobe Photoshop, using the standard paintbrush, masking, and selection tools. A trained artist can edit a texture map in less than three minutes.

Figure 10. The system fabricates the completed customized CG figurine (M_{RF}) with a customized texture map (T_{RF}), using a color 3D printer. We use a ProJet 660Pro.

Fabrication

Any 3D printer that can handle manifold triangulated meshes will work with our system. We use the state-of-the-art ProJet 660Pro (www.3dsystems.com/3d-printers/professional/projet-660pro#.UlNaEbx9WQU), which employs a composite powder as the printing material. It has a resolution of 600 × 540 dpi and a printing volume of 254 × 381 × 203 mm, and can resolve features as small as 0.1 mm. After printing, a range of finishing options are available to increase durability and provide different aesthetics. Figure 10 shows a printed Belle figurine.

Our system prints figurines from 76.2 to 177.8 mm high. Smaller figurines have fragile limbs and insufficient facial detail to be easily recognizable. The printing throughput varies considerably and depends on the figurine size and model. It ranges from an approximate maximum of six figurines per hour for the 76.2-mm figurines (48 figurines per batch) to a minimum of 0.75 figurines per hour for wide 177.8-mm figurines (six figurines per batch). After printing, we apply a matte resin finish to increase durability.

Figure 11. Digital figurines our system created. The artist applied different aesthetic treatments during color matching. To reflect different ages, the male figurines have bodies of different heights, whereas the female figurines have different chest-to-waist ratios.

Deployment

While deploying our system for testing and commercialization, we've fabricated thousands of figurines encompassing 20 models and five color-matching protocols. We've also experimented with creating different body types to reflect the customer's age. Four body types are available for the princess figurines. Figure 11 shows figurines with different aesthetics and age-adjusted body types.

Disney calls the commercial deployment of this technology the "D-Tech Me experience." World of Disney, a store at Walt Disney World Resort in Orlando, Florida, hosted D-Tech Me: Disney Princess. Hollywood Studios, also at Walt Disney World Resort, and Star Wars Celebrations VI, an annual gathering of *Star Wars* fans, hosted D-Tech Me: Carbon Freeze Me.

During the experience, customers got 3D-scanned in a setup themed according to the chosen D-Tech Me mode. They were scanned a few times and then chose their favorite portrait based on the scanned images. The system reconstructed the selection, and the scanner operator showed the customers a low-resolution version of the 3D facial scan, which often generated excitement. D-Tech Me: Disney Princess customers could choose from seven princesses and many hair tones. The color-matching artist selected the skin tone; the scanner operator selected the body type. D-Tech Me: Carbon Freeze Me offered a single figurine representing Han Solo frozen in carbonite.

The rest of the process was transparent to the customers, who received their customized figurine six weeks later by mail. Once the order was completed, the system performed a full 3D reconstruction of the image data collected by the scanner to create the customer's facial 3D scan. Reconstruction took up to 20 minutes on a standard computer.

The system sent the face scan to a workstation, where a human operator annotated the feature points required for registration. A trained operator could do the job in less than a minute.

After annotation, the system automatically performed registration and attachment. The corresponding computation time was approximately 50 seconds on a standard computer. Then, the system sent the assembled digital figurine to the artist's workstation for color matching. Editing the texture map took approximately three minutes. The actual time depended on not only the artist's skill but also the figurine's aesthetic requirements. Processing a princess figurine took longer than processing a carbonite one. Finally, the system sent the completed figurine to the 3D printer for fabrication, followed by finishing, packaging, and shipping.

Our system's commercial deployment and the production of thousands of figurines attest to its robustness and efficacy. As an additional bonus, the system has engaged the general public with state-of-the-art computer vision, graphics, and 3D-fabrication technology in an intuitive and tangible manner. We believe that as 3D printing technology becomes more prevalent, such customized production systems will become more common and have a broad impact, beyond the merchandise domain we explored here.

Although our system is automated enough to allow for high throughput, a fully automated system is our goal. As 3D printing technologies evolve, resolution will increase while fabrication time decreases. These changes will shift the bottleneck of the system's throughput from fabrication to manual labor. Accordingly, moving to a fully automated system will become more important.

Current 3D facial registration still requires human interaction for initialization when high precision is required and no constraints are placed on the target face's expression. So, one of our goals is fully automatic facial registration. We'd also like to further automate color matching by performing feature extraction, segmentation, and expression analysis on the texture map. This would let the system perform aesthetic editing without human intervention. However, the system would then need algorithms tailored for each figurine model's aesthetics, increasing the system's complexity.

Many questions remain unanswered regarding the self-recognition process that motivates a customer to identify with the fabricated figurine. Our current system ignores the customer's full body shape and hairstyle; however, these elements are important to many individuals' self-perception. We plan to explore techniques that let figurine fabrication incorporate the customer's body and hair. This issue is not only about developing the right technology but also about customer preference and self-perception. Finding the right balance between preserving the customer's hairstyle and body shape and being truthful to the style and character depicted by the figurine won't be trivial. It also might well be a matter of personal preference that must be adjusted for each customer.

Acknowledgments

Our system's commercial deployment was a joint effort across several organizations in the Walt Disney Company. We acknowledge Tom Ngo, Cyd Tetro, and Johnnie Rush for producing the project and the rest of the staff at Disney Research, Disney Theme Park Merchandise, and the New Technology Group who worked on the project throughout its development. [io]development performed the software-engineering work for deployment; 3D Systems provided 3D-printing hardware and expertise during development and deployment. Patent pending, application US 14/474,625.

References

1. L. Hvam, N.H. Mortensen, and J. Riis, *Product Customization*, Springer, 2008.
2. T. Beeler et al., "High-Quality Single-Shot Capture of Facial Geometry," *ACM Trans. Graphics*, vol. 29, no. 4, 2010, article 40.
3. J.R. Tena et al., "A Validated Method for Dense Non-rigid 3D Face Registration," *Proc. IEEE Int'l Conf. Video and Signal Based Surveillance* (AVSS 06), IEEE, 2006.
4. F. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, 1989, pp. 567–585.
5. E. Weisstein, "Barycentric Coordinates," *MathWorld*, Wolfram, 2007; http://mathworld.wolfram.com/BarycentricCoordinates.html.
6. G. Taubin, "Curve and Surface Smoothing without Shrinkage," *Proc. IEEE Int'l Conf. Computer Vision* (ICCV 95), IEEE, 1995, pp. 852–857.
7. I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis*, John Wiley & Sons, 2002.
8. M. Botsch and O. Sorkine, "On Linear Variational Surface Deformation Methods," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 1, 2008, pp. 213–230.
9. O. Sorkine et al., "Laplacian Surface Editing," *Proc. 2004 Eurographics / ACM Siggraph Symp. Geometry Processing* (SGP 04), ACM, 2004, pp. 175–184.
10. Y. Yu et al., "Mesh Editing with Poisson-Based Gradient Field Manipulation," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 644–651.
11. J. Vollmer, R. Mencl, and H. Muller, "Improved Laplacian Smoothing of Noisy Surface Meshes," *Computer Graphics Forum*, vol. 18, no. 3, 1999, pp. 131–138.
12. P. Pérez, M. Gangnet, and A. Blake, "Poisson Image Editing," *ACM Trans. Graphics*, vol. 22, no. 3, 2003, pp. 313–318.

Many questions remain unanswered regarding the self-recognition process that motivates a customer to identify with the fabricated figurine.

J. Rafael Tena *is a research engineer at Disney Research in Pittsburgh. His research interests center on the human face, particularly facial modeling and animation, facial recognition and perception, and facial motion and retargeting. Tena received a PhD in computer vision and graphics from the University of Surrey. He's a member of IEEE and ACM. Contact him at rafael.tena@disneyresearch.com.*

Moshe Mahler *is an artist and modeler at Disney Research in Pittsburgh. He received his BFA from Carnegie Mellon University. His artworks and animated films have been featured in a number of venues internationally, including Siggraph 2005, 2007, and 2011. Contact him at moshe.mahler@disneyresearch.com.*

Thabo Beeler *heads the group for capture and effects at Disney Research in Zurich. His research interests center on the accurate digital reproduction of human faces. Beeler received a PhD from ETH Zurich. He's a member of ACM and Eurographics. Contact him at thabo.beeler@disneyresearch.com.*

Max Grosse *is a software engineer at Disney Research in Zurich. His interests include software design and engineering, especially integration, adaption, and deployment of research code. Grosse received a BSc in media systems from Bauhaus Universität Weimar. He's a member of ACM. Contact him at max.grosse@disneyresearch.com.*

Hengchin Yeh *is a PhD candidate in the Computer Science Department at the University of North Carolina at Chapel Hill. He previously was a research intern at Disney Research. His research interests are physically based sound simulation for graphics applications, including sound synthesis and sound propagation. Yeh received a master's in computer science and information engineering from National Taiwan University. Contact him at hyeh@cs.unc.edu.*

Iain Matthews *is a principal research scientist and the associate research director at Disney Research in Pittsburgh. He holds an adjunct faculty appointment at Carnegie Mellon University's Robotics Institute. His research interests are computer vision, human-computer interaction, and speech generation and recognition. Matthews received a PhD in computer vision from the University of East Anglia. He's a member of the Institution of Engineering and Technology, IEEE, and ACM. Contact him at iainm@disneyresearch.com.*