Drag, Drop, and Clone: An Interactive Interface for Surface Composition

Ryan Schmidt
University of Toronto

Karan Singh
University of Toronto

Abstract

We describe a novel interface for composition of polygonal meshes based around two artist-oriented tools: Geometry Drag-and-Drop and Mesh Clone Brush. Our drag-and-drop interface allows a complex surface part to be selected and interactively dragged to a new location. We automatically fill the hole left behind and smoothly deform the part to conform to the target surface. The artist may increase the boundary rigidity of this deformation, in which case a fair transition surface is automatically computed. Our clone brush allows for transfer of surface details with precise spatial control. These tools support an interaction style that has not previously been demonstrated for 3D surfaces, allowing detailed 3D models to be quickly assembled from arbitrary input meshes. We evaluated this interface by distributing a basic tool to computer graphics hobbyists and professionals, and based on their feedback, describe potential workflows which could utilize our techniques.

Keywords: modeling interfaces, 3D drag-and-drop, clone brush

1 Introduction

Interactive tools for conceptual 3D design continues to be a fertile area of research in computer graphics and human computer interaction. 3D surface composition is an important problem in this area, as designs often evolve from a combination of existing ideas and models. In this paper, we explore interactive techniques for shape reuse and composition in 3D modeling.

Recent developments in shape composition have attempted to address the problem in two ways, which we will call *part fusion* and *detail transfer*. In part fusion, arbitrary surface parts are cut from one model and attached to another. To overcome limitations of early volumetric approaches, recent methods rely on open boundary loops on the part and target surfaces, which can be automatically aligned [Funkhouser et al. 2004; Sharf et al. 2006] and joined via fair surfaces or global deformation of the part [Sorkine et al. 2004; Yu et al. 2004]. In contrast, detail transfer techniques utilize planar parameterizations to establish correspondences between source and target *areas*, then transfer displacement [Biermann et al. 2002] or differential [Sorkine et al. 2004] information.

The focus of these works is on the algorithmic and geometric aspects of shape composition, and the interfaces provided, though in many cases straightforward to use, often involve operations that would make design exploration inefficient. For example, part fusion techniques require a compatible hole to be cut in the target surface. If the artist is dissatisfied with the result, another hole must be created. This workflow clearly lacks the simplicity and efficiency of similar interactions found in 2D image editors, where an artist can simply drag selected pixels from one location to another. Similarly, detail transfer techniques have focused on "cut-and-paste" edits, copying entire enclosed regions. Examining 2D image editors, we note that a much more powerful and efficient interaction is the *clone brush*, an intuitive tool which allows the artist to *selectively* transfer details between the corresponding areas.

Inspired by these 2D interactions, we have designed two novel tools for composition of arbitrary surface meshes. To perform fusion-style tasks for boundary-based features, we present *Geometry Drag-and-Drop*. This technique allows an artist to select an arbitrary part and drag it along the surface to a new location, or onto another surface, while automatically filling the hole left behind. For detail transfer we introduce the *Mesh Clone Brush*, which allows the artist to paint directly onto the surface to specify the extent and intensity of the transferred details. Both tools operate in real-time, providing instant feedback and opportunity for refinement.

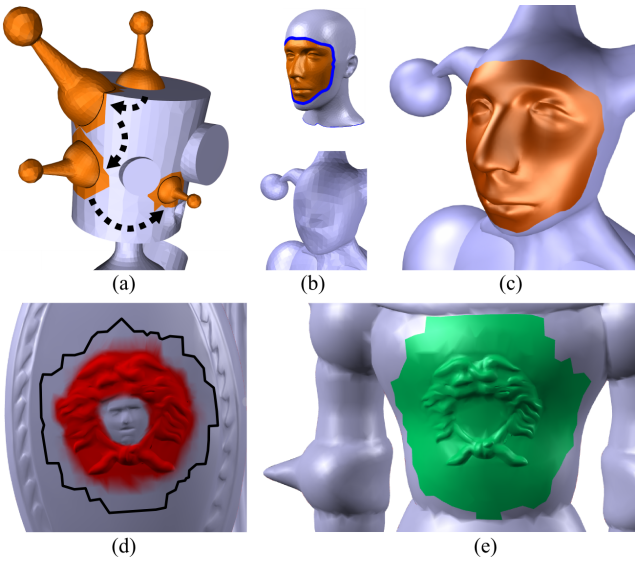

Figure 1: *We present a novel artist-oriented interface for surface composition. Arbitrary mesh parts can be dragged-and-dropped from either one location on a surface to another (a), or between two completely different surfaces (b,c). In either case, the hole left behind is automatically filled. Our mesh clone brush supports precise control over the extent of detail transfer tasks by painting directly onto the mesh surface (d,e).*

The fluidity and efficiency of our approach has has not been available in any previous system. Hence, we evaluated Geometry Drag-and-Drop by distributing a simple tool to hundreds of artists, hobbyists, and 3D professionals. In Section 4 we distill the feedback gathered during this experience into a set of *workflows* encompassing tasks common to different user groups, and explain how our techniques can be applied to simplify these workflows.

1.1 Related Work

Many recent works in shape modeling have placed a priority on *intuitive interactions*, such as those in feature-preserving deformation [Botsch and Sorkine 2008; Gal et al. 2009], freeform surface representations [Nealen et al. 2007], shading-based editing [Gingold and Zorin 2008], and image-guided modeling [Gingold et al. 2009]. Comparatively fewer works in shape composition have considered interaction issues, most intead focusing on algorithmic aspects such as automatic alignment and blending. Here we focus our review on works which specifically address interactive aspects of creating a new shape given specific parts or source models.

Part Composition If the models to combine belong to a well-defined class, such as humanoid bipeds, pre-computed compatible segmentations allow specific parts to be interchanged via single clicks, as in Shuffler [Kraevoy et al. 2007]. The Modeling by Example system [Funkhouser et al. 2004] provides a more general interface, where the artist can select parts of any models, then have them automatically aligned and joined. SnapPaste [Sharf et al. 2006] introduces the intuitive notion of *snapping* into this workflow, providing the artist with guided alignment based on the current spatial orientation of the part. A key aspect of the user experience is how the shapes are merged. If the boundary of the part is relatively similar to the target surface, simple blending may suffice [Kanai et al. 1999; Sharf et al. 2006]. For more complex joins, variational techniques can automatically compute detail-preserving part deformations [Sorkine et al. 2004; Yu et al. 2004; Fu et al. 2007] and be combined with part boundary optimization to further improve results [Hassner et al. 2005; Huang et al. 2007].

Detail Transfer Unless the target surface and boundary loop are flat enough to use a planar projection [Yu et al. 2004], the techniques mentioned above require a suitable boundary loop or hole on the target surface to be manually specified. The main implication of this step is that if the artist wishes to change the composition, previous holes must be removed and new ones added. In addition, the part cutting step leaves a hole in the surface which may need to be filled if the source surface is to be re-used. If the source and target models are represented by multiresolution hierarchies, then holes can be avoided altogether. Instead *compatible parameterizations* allow details to be subtracted from the source and added to the target. Biermann et al [2002] describe techniques for interactively cutting and pasting multiresolution surfaces. This interface can be extended to arbitrary meshes by either explicitly constructing a base surface at the source [Fu et al. 2004], or by transferring differential information via the parameterization [Sorkine et al. 2004].

Drag and Drop Procedural surface hierarchies [Barghiel et al. 1995; Schmidt and Singh 2008] intrinsically support intuitive drag-and-drop manipulation of geometry, although they are limited to displacement-style features. Our robust handling of arbitrary parts would benefit these frameworks, and our interface could even be used to decompose a static mesh into layered semantic parts. Finally, Suzuki et al [2000] directly address the drag-and-drop problem for small rigid parts of triangle meshes, by locally remeshing around the boundary of the part as it translates through 3D space.

2 User Interface

In this section we describe our interface from the artist's perspective. The technical details underpinning each technique are described in Section 3. We also briefly mention various control parameters that are provided to the artist. We stress that these increase expressivity, but are not strictly necessary - aesthetically appealing results are generated automatically by the default settings.

2.1 Selection and Part Extraction

The first step of a drag-and-drop operation is to select a region-of-interest on the 3D surface. Our interface supports two selection tools. In the first the artist can draw one or more connected strokes on the surface, from multiple viewpoints, to select an interior set of connected faces. Alternately a lasso stroke beginning on the scene background cuts through the model and selects the enclosed faces. *Intelligent scissor* techniques [Funkhouser et al. 2004] for efficient part selection could be applied to here, although in practice we observe that our selection boundaries often lie some distance away from geometrically-salient features (Fig. 2).

Once a selection is made and a drag operation initiated, the selection interior is separated from the base mesh into a *part*, and the resulting hole is filled with a patch that smoothly blends with the hole boundary. The artist may use simple controls to manipulate the shape of the fill region (Fig. 2), and once satisfied, accept the current result and move on to the drag-and-drop phase.

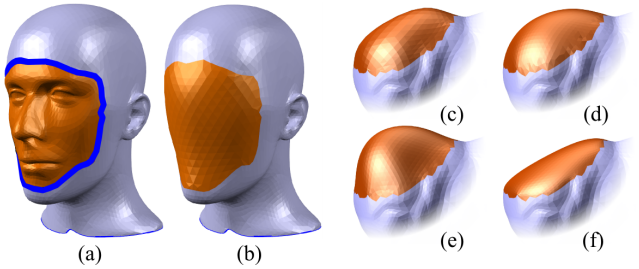

Figure 2: *When the artist initiates a drag operation, our interface extracts the selected surface region (a) and smoothly fills the hole left behind (b). Two simple parameters are provided to support reshaping of the fill surface, if desired (e-f).*

2.2 Geometry Drag-And-Drop

While in the drag-and-drop state, the artist can interactively drag the active part across the surface, uniformly scale it, and rotate it's boundary "in the surface" (Fig. 1). These constrained interactions do away with the challenges of 3D manipulation, allowing for efficient part placement. The part mesh is deformed such that its boundary is embedded in the target surface, and a seamless connection between the target surface and part is displayed, providing a live preview of the result of the drop operation. If the part contains too much goemetric detail to preview interactively, we show a reduced-resolution version during interaciton, and compute the full-resolution result when the drop is completed.

To support addition re-use of parts, our interactive tool includes a *part library*, a scrolling list attached to the modeling window. The artist can drag the active part into the part library, storing it for later use. When a part from the library is dragged onto the active model, we skip the extraction state and directly enter drag-and-drop mode.

2.3 Variable Part Rigidity

By default, our drag-and-drop operation deforms the part to conform to the rigid target surface. An alternative is to smoothly deform the target region to fit the part. One can then easily imagine a continuum of solutions between these two extremes, with a partial deformation of both the part and target region.

We provide just such a capability. The artist is given a *rigidity* control over the part boundary, and as rigidity is increased above 0, the part boundary smoothly blends from the embedding in the target surface to its initial rigid configuration (oriented with respect to the target location). In this case a smooth, variable-radius transition surface is computed between the target region and the part. Extending this smooth connection, we also provide provide the artist with the ability to offset the part into or away from the target surface, increasing the expressive capability of the tool. Natural extensions would be to allow for additional rigid transformation of the part boundary, or even interactive deformation of the loop vertices.

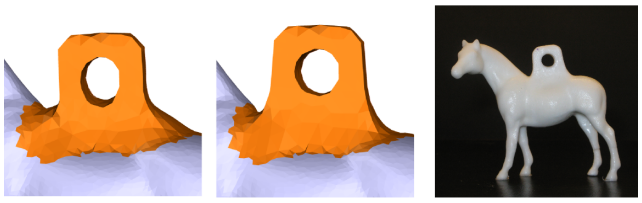

Figure 3: *Using a global deformation to fit a part to a target surface can distort important shape semantics (a). Increasing boundary rigidity preserves the interior shape (b), allowing a precise physical part to be manufactured (c).*

2.4 Mesh Clone Brush

A clone action involves dynamically copying from a source to target region, where the selection extent and opacity is determined by brushing. For our 3D mesh clone brush, we copy detail vectors, found via an automatically-generated base surface (Fig. 1). To initiate a clone action, the artist marks the source and target regions with geodesic discs positioned by dragging their center points across the surface, with geodesic radius controlled by the mouse wheel. A preview of the smoothed base surface is shown to give the artist a sense of the details that will be transferred. Once both areas are specified, brushing can begin.

With the standard 2D clone brush the artist implicitly defines the target area with the first paint stroke. However, unlike in 2D images, two patches on a surface do not have a meaningful coordinate-system correspondence, hence the need to mark the target area before brushing. Similarly, the non-obvious correspondence between source and target region leads us to allow painting strokes to be drawn on either. Since the target region may need adjustment, the currently painted details can be dragged across the surface, as well as scaled and re-oriented, as in our drag-and-drop tool. The smoothness of the source base surface can also be manipulated at any time. During each of these interactions, a live preview of the clone operation is shown in the target region.

3 Algorithms

Our interaction techniques could be implemented using many different algorithms. We have focused on linear variational methods because they are simple to implement and highly efficient. For part deformation we also provide a novel geometric technique, the COILS deformer, which, though non-smooth, provides reasonable solutions in cases that are problematic for linear variational approaches. Details of this method are provided in Appendix B.

3.1 Filling Holes

To implement our part separation (Section 2.1) we must fill the hole left behind. As it could include interior holes or topological handles, the part mesh is of no use in solving this problem. Although excellent techniques for completely automatic hole filling have been developed [Davis et al. 2002; Liepa 2003], which can even copy local context [Sharf et al. 2004], we explored a deformation-based approach, with optional clone-brush post-processing, because it allows us to provide the artist with some control over the result. Hence, we construct a planar mesh patch and deform it to fit the hole using rotation-invariant coordinates (RIC) [Lipman et al. 2005] (Fig. 4). Given boundary orientation and positional constraints, RIC first solves for an orientation field, and then computes

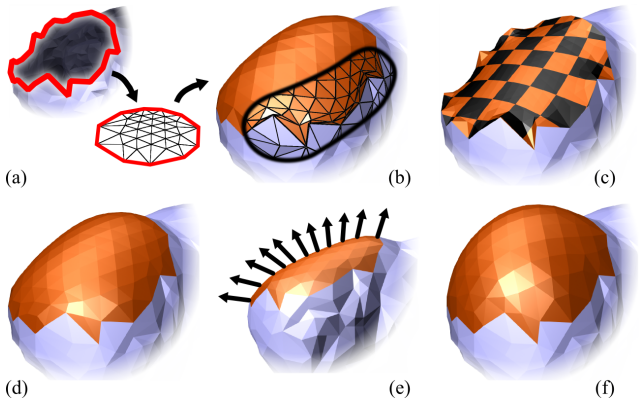

Figure 4: *A hole can be filled by embedding the boundary loop in a planar disc (a) and then deforming the disc, but this leads to undesirable triangle distortion (b). Parameterizing a membrane fill surface (c) creates a planar mesh with a boundary that matches the hole. This mesh produces a fill with smooth boundary (d) but the interior normals deviate from the estimated orientation field (e). Optimizing the scale of the planar mesh improves the result (f).*

a Laplacian deformation [Sorkine et al. 2004] using the orientation field to transform the differential vectors.

This deformation approach is sensitive to the shape of the planar mesh - it's boundary should "fit" the 3D hole boundary. Since this mapping is unknown, we first embed the 3D boundary in a 2D circle, and generate a regular interior mesh using constrained Delaunay triangulation (CDT) [Shewchuk 1996]. This mesh is deformed to fit the 3D hole, but triangles around the boundary are often highly distorted (Fig. 4b). So, we parameterize this smooth 3D fill mesh using a free-boundary conformal map [Desbrun et al. 2002], creating a 2D boundary which "looks like" the 3D boundary. Finally we generate a regular interior mesh, again using CDT, followed by sliver removal and uniform Laplacian fairing.

This new patch is then deformed into 3D, resulting in a smooth but flattened fill surface, whose normals deviate significantly from the desired normals generated by RIC. This occurs because the planar mesh has the scale of the membrane surface, so to allow it to fit the rounder normal field, we must scale the differential coordintes. To find this scaling factor, we solve a simple 1D optimization problem:

$$\arg\min_{s} \sum_{i} |1 - N_i \cdot \mathbf{n}_i(s)| \qquad (1)$$

where N_i is the frame normal determined by the orientation step of RIC, and $\mathbf{n}_i(s)$ is the mesh normal, estimated after solving the position step of RIC with the differential vectors scaled by s. Since we can pre-factor the system matrix and compute solutions for a different s with only a back-substitution [Botsch et al. 2007], this optimization is very efficient, and generally converges in 10-20 steps. After optimization, allowing the artist to manipulate this scaling parameter gives a useful control over the "flatness" of the fill surface.

3.2 Part Insertion

To transport a part across the surface, we use an improved version of the Discrete Exponential Map (DEM), described in Appendix A. This technique parameterizes a geodesic disc around a central *seed point*. After the hole surface is filled, we find the (approximate) geodesic center and parameterize a region large enough to contain the part boundary. We then embed the boundary in the parameterization, and also store relative tangent-normal frames at each bound-

ary vertex. Given a second DEM parameterization, we project the embedded *uv*-boundary to the new surface (Fig. 5), and then transfer the relative frames back to global coordinates using the new surface tangent-normal frames. Having determined the necessary boundary constraints for rotation-invariant coordinates (RIC), the part mesh is deformed form its original shape and orientation to a configuration which precisely fits the new location. Once the part is deformed, we stitch it into the target mesh by inserting the boundary within the DEM parameter space.

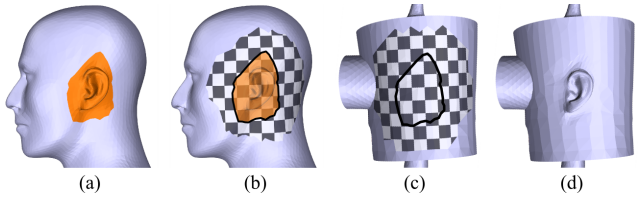

Figure 5: *When a selected region (a) is separated into a part, we parameterize the underlying fill surface (b) and embed the part boundary in the parameterization. Given a parameterization of a new surface, we can transfer the boundary (c) and then compute a shape-preserving deformation of the interior (d).*

Again, straightforward application of RIC produces undesirable results when the boundary is significantly deformed (Fig. 6a). In this case the new mesh normals do not deviate significantly from the desired RIC-estimated normals, but rather the mesh is simply too deformed, and so we must manipulate the boundary orientation constraints. We can rotate the constraint frame at each boundary point $\mathbf{p}_i$ around the axis $(\mathbf{p}_{i+1} - \mathbf{p}_{i-1})$, then find the new solution via back-substitution. The optimal global rotation angle θ is found by minimizing the total area deformation of the mesh triangles:

$$\arg\min_{\theta} \sum_i \left[\left(A_i^{\text{orig}} / A_i^{\text{def}(\theta)} - 1 \right)^2 A_i^{\text{orig}} \right] \qquad (2)$$

where A_i is the area of triangle T_i in the original or deformed configuration. During interaction we use a time-constrained 1D line-search to solve this problem, ensuring real-time feedback, and then fine-tune during idle time. Optimizing for per-vertex angles can further refine the deformation, but is too expensive for interactive use. The net result of boundary angle optimization is a significant improvement in the deformed shape, but at the cost of a smooth transition at the shape boundary (Fig. 6b). Hence, we allow the artist to manipulate this boundary angle, and also the global differential scaling factor, to interactively tune the part shape.

We have also developed the COILS deformer (Appendix B), a plug-in alternative to RIC which, for the specific boundary-value problem we are solving, produces results comparable to differential techniques (Fig. 6). The same boundary constraints are applied, and a similar boundary-frame optimization is necessary, although for COILS the area change is not a suitable objective. Instead we minimize the deviation between the boundary constraints and mesh normals along the boundary, effectively optimizing for boundary continuity. This is possible while preserving interior rigidity for extreme deformations because COILS concentrates deformation near the boundary, rather than distributing it smoothly as in RIC. This geometric property is particularly effective when interior holes are present. As with the method of Sumner et al [2007], the COILS deformer can also be applied to non-manifold or disconnected parts. However, this flexibility comes at a significant cost - the COILS deformer is non-smooth, and this can lead to visual artifacts on high-resolution smooth regions of the surface. As a result, neither technique has a clear advantage, and as they are roughly equivalent in efficiency, we provide the artist with the ability to select either depending on the task at hand.

This approach to part drag-and-drop is quite expensive. We found that feedback rates can be made more consistent by simplifying the part interior via edge collapses until we have reduced it to 1500 vertices, and then representing the full-resolution part as an offset surface. Considering Fig. 9, we can drag the head part (22k vertices) over a target area containing 1k vertices at several frames per second, While a part with 10k vertices runs at 10-15 frames per second. To support parts with extreme levels of detail, we could compute a multiresolution hierarchy [Kobbelt et al. 1998] and only construct the non-interactive detail levels after the part is dropped.

3.3 Variable Boundary Rigidity

The previous technique deforms the part boundary to to conform to the target surface. This can result in extensive deviation from the initial shape, which may be undesirable. Hence, we provide the option of a boundary with variable rigidity, implemented using a linear variational curve deformation.

We adapt the Laplacian mesh optimization technique described by Nealen et al [Nealen et al. 2006] to curves. In this method, the differential vectors are set to 0 and a weighted "soft" positional constraint is applied to each vertex, with the weight effectively controlling an interpolation between complete fairness and the original shape. We would instead like to blend between the original boundary loop B_o, and the 3D embedding on the target surface, B_t.

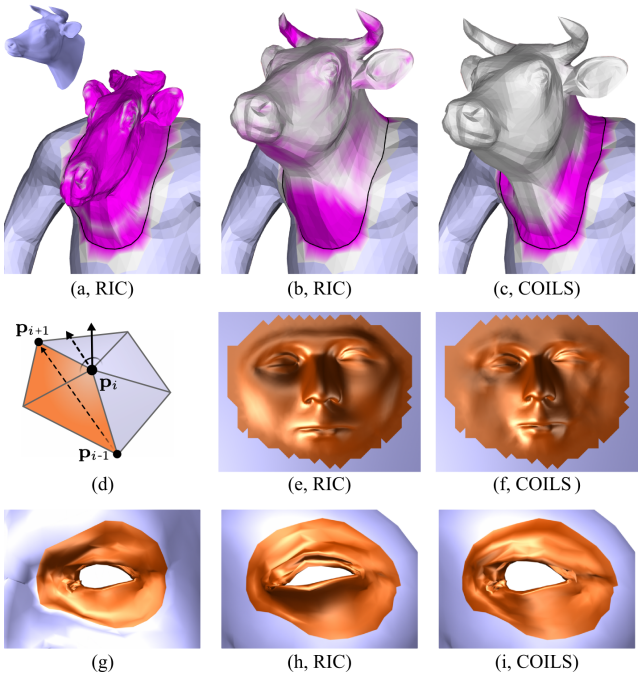

Figure 6: *Dropping a cow part onto a torso while maintaining boundary continuity causes involves too severe of a bend for rotation-invariant coordinates (a). Optimizing boundary frames (d) to minimize distortion (purple) reduces boundary continuity but preserves interior shape (b). The COILS deformer concentrates distortion near the boundary, allowing for continuity with more interior rigidity than RIC. The least-squares optimality properties of RIC allow it to outperform the non-smooth COILS deformer on high-resolution surfaces (e,f), while the COILS deformer is more robust at preserving the shape of interior holes (g-i).*

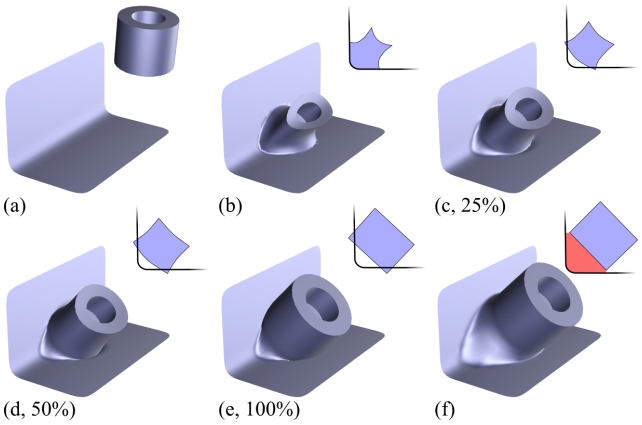

Figure 7: *Embedding the boundary of a cylindrical part on a corner surface (a) results in extreme distortion (b), which can be reduced by increasing boundary rigidity (c-e). Offsetting the boundary from the surface and increasing the radius of the fair transition region produces a useful socket (f).*

Hence, we compute the differential vectors using B_o and constrain the vertex positions using B_t. Note that as the differential vectors are not rotation-invariant, we must compute a suitable transformation which aligns B_o with B_t [Horn 1987].

With a very low weight, these positional constraints only fix the global position of the solution, which otherwise looks like B_o. As the weight increases, the curve vertices are smoothly deformed towards B_t. The rigidity parameter we provide the artist is simply the inverse of this weight. With this semi-rigid boundary, the deformed part will be disconnected from the target surface, so we find a fair transition surface by solving a linear variational thin-plate problem [Botsch and Sorkine 2008] within a variable-radius cylindrical region of the target surface. Given this machinery, it is straightforward to allow the boundary to be shifted in to or out of the surface.

3.4 Local Displacement Transfer

To implement our clone brush, we must transfer detail information from the source to target region. We extract the details by computing a smooth base surface in the source region, and then find per-vertex displacement vectors by subtracting the smoothed positions from the originals. The base surface is found using the Laplacian mesh optimization technique [Nealen et al. 2006], adding soft constraints for each interior vertex. We allow the artist to control the scale of smoothing via the weights on these interior constraints. Once the smoothed base and target regions are defined, we define a mapping between them by computing a local DEM parameterization for each. As the artist paints on the source vertices, we transfer the affected triangles via parameter space to the target region, stitch them in, project back to the target surface, and apply the detail vectors to the inserted vertices.

Re-meshing in the target region can be avoided by sampling the displacement field and offseting the existing vertices, but salient geometric details can easily be lost if the target region has insufficient resolution. This is likely to be frequent if one considers the practical use cases for a clone brush tool. Another alternative is to transfer details using linear differential methods [Sorkine et al. 2004], avoiding the need to compute a smooth base surface. As the source region-of-interest can change with each brush stroke, with this approach the system matrix must be frequently updated, ruling out the high interactivity artists expect from brush-based tools.

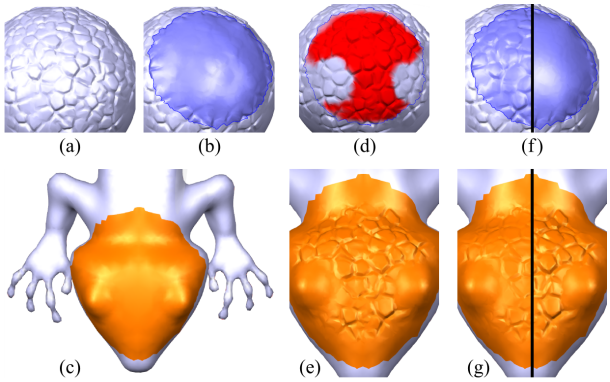

Figure 8: *To clone detail (a) the artist first selects an automatically-smoothed source region (b), then marks a target area (c). During painting on the source (d) the target is interactively updated (e). The level of smoothing, and hence the intensity of transferred detils, can be varied during painting (f,g).*

4 Evaluation and Applications

In its simplest form, our drag-and-drop composition technique combines surface-constrained part positioning, global deformation, and geometry merging into an atomic action controlled by a single parameter: the position of mouse cursor over the 3D surface. Even casual computer users with no 3D modeling experience have been able to immediately use this tool. Similarly, those familiar with the 2D pixel clone brush have understood our 3D counterpart in moments. Based on these experiences, we claim success in creating a simple and efficient interface for 3D shape composition.

However, in creating such a simple interface we have also significantly constrained part positioning, deformation, and merging. While constrained shape composition has clear utility in restricted environments such as the Spore creature creator [Electronic Arts 2009], it was less clear whether our tool would be useful to professional practitioners of 3D modeling. Evaluation of this sort is quite difficult, and Greenberg and Buxton [Greenberg and Buxton 2008] note the dangers of applying premature usability evaluation to interface prototypes. Hence, rather than a limited formal comparison, we performed a large-scale informal evaluation to solicit the widest possible feedback.

We distributed a basic version of our drag-and-drop tool to a wide community of computer graphics hobbyists and professionals over a 2 month period. Minimal instructional material was provided (a 2-minute demonstration video and a 4-minute tutorial video with text captions). Although we offered technical support via e-mail and an online forum, most requests involved advanced uses of our tool or software bugs. We have yet to encounter a user who was confused about the basic operation of our drag-and-drop interface.

The response to our tool was overwhelmingly positive. We received feedback from a wide range of 3D practitioners, who wished to use our tool in many more ways than we had imagined. One aspect of this feedback which was particularly interesting was the variation in *workflows* and diversity of applications that formed the sets of recurring tasks for different user groups. In the following paragraphs we describe how our interactive composition techniques could be integrated into some of these workflows.

Part Assembly One area of interest in our interface was from special effects artists who assemble models based on 3D scan data. Complex objects are often scanned in parts, with varying levels of

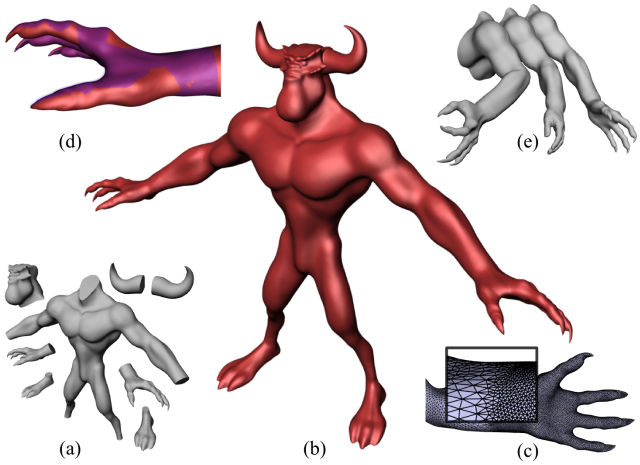

Figure 9: *We assembled 8 high-resolution components (a) into a manifold surface (b), spending 5-10 minutes per part. Stitching between varying levels of detail (c) was handled automatically, and we achieved good alignment with the original parts (purple) without difficulty (d). We can re-orient or replace parts using smooth transitions, and even re-pose the model by cutting at joints (e).*

resolution used for different regions. Assembly of these scans into a consistent surface is a tedious process, particularly if the object is a human who moves between scans. A professional modeling studio provided us with a sample task, shown in Fig. 9, which we quickly completed using our tool. In addition to the "correct" assembly, our smooth transitions allowed parts to be positioned at novel orientations, or replaced entirely. By cutting at joints it was even possible to re-pose the model, while avoiding the loss of volume often observed with deformation-based reposing.

A related problem is that of assembling specific models from part catalogs, in particular to provide customized human models. In this case the parts must be deformed to conform to the new surface, which our approach automates. In addition, the complexity of current tools means that the customer must pick parts from images and then specify design changes to an artist. With our interface, the customer can efficiently experiment with different character designs, reducing the need for expert guidance (Fig. 10).

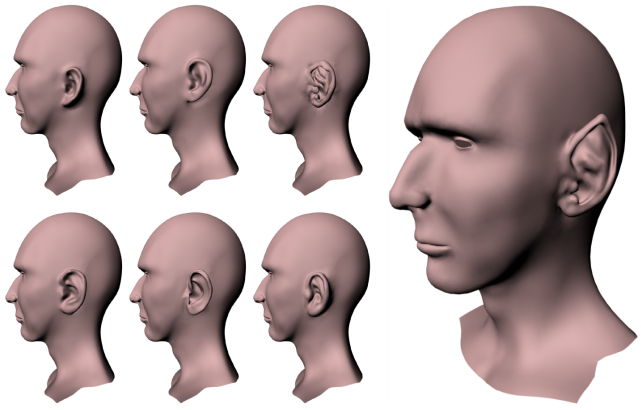

Figure 10: *Each ear on the left took only a few seconds to position, making it efficient to experiment with design variations. The top-right model was imported into a brush-based sculpting application and used as the basis for further exploration (right).*

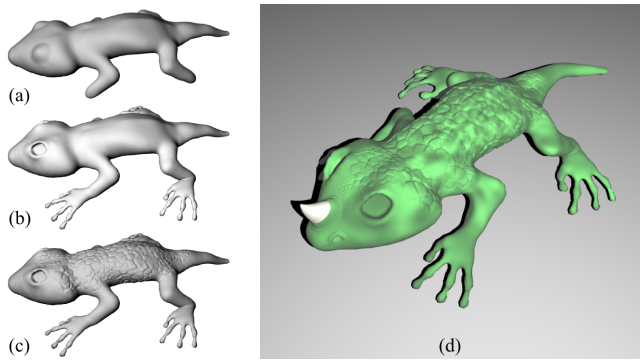

Figure 11: *To mock up a lizard model, we first sculpted a sphere into a suitable body shape (a) then transferred the more complex features from other models (b). Next we used our clone brush to copy from a rough, scale-like surface (c), then added a few more parts and rendered with a procedural texture (d). No more than 10 minutes was spent on each step, but the final model would suffice for most digital mock-up purposes.*

Rapid Base-Mesh Creation 3D sculpting tools are used to paint extreme levels of detail onto simplified "base-meshes". Creating the necessary base-mesh for a sculpting task can be time consuming, and often 3D sculptors resort to stock models which must then be deformed or otherwise edited into a more suitable form. With our tool, a library of stock parts can be quickly assembled into a much wider range of base-meshes suitable for further detailing (Fig. 10, 11). The efficiency of our interface also makes it practical to drop in stock parts as needed, or even to remove and replace an existing part when a design variation proves unsuccessful.

Model Pre-Visualization A growing area of application for computer graphics in film and entertainment industries is *pre-visualization* (or *previz*). 3D mock-ups with varying levels of fidelity are now created for many digital and live-action shots, and used for virtually every production task, from basic staging and art direction to detailed camera, lighting, and rendering setup. Creating suitable models for higher-fidelity previz using digital sculpting tools is a time consuming and expensive process, as the models will ultimately be discarded. Several 3D artists noted that our composition tools would make modeling for previz much more efficient. We explore this use case in Fig. 11.

Satisfying Design Constraints A wide range of factors are taken into account during design processes, from art direction and marketing requests to immutable engineering issues. Hence, a task common to a wide range of digital modeling workflows is updating existing models to satisfy new design constraints. These updates often involve relatively minor changes, but even a simple shift of a gas door on a car can take several hours to accomplish with traditional modeling interfaces. Using drag-and-drop, however, a professional modeler can accomplish such tasks in seconds (Fig. 12).

Scan Repair Perhaps the most enthusiastic testers of our tool were artists who must work with meshed scan data. 3D scans rarely capture all the necessary surface detail, and often occlusion or motion results in holes and incorrect forms. Cleaning up this data is a tedious manual process. Users told us that simply selecting and dragging off a hole, leaving a fair surface behind, was a major improvement to their workflow. Dragging in or cloning detail from regions where the scan is more accurate, or from other models, allows invalid regions to be efficiently repaired (Fig. 13).

Rapid Prototyping 3D printing has become relatively commonplace in many areas of industrial design, however realizing a 3D

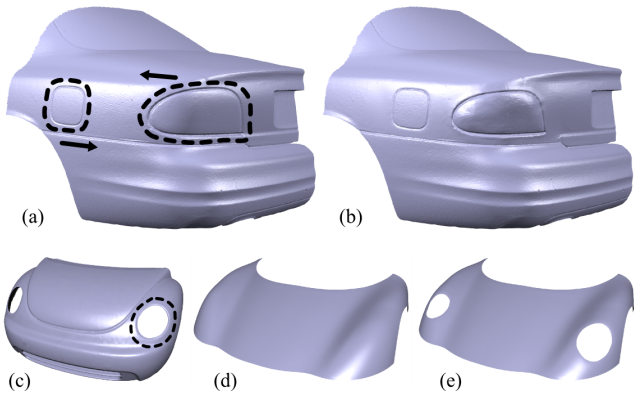

Figure 12: *Features of a scanned car surface are repositioned via drag-and-drop to satisfy new engineering requirements (a,b). In (c-e), we transfer a design stylization by dragging headlight portholes from one car body to another.*

model as a physical object introduces its own set of particular challenges. For example, a model may need to be significantly modified so that parts can actually be assembled after printing. A related problem is that many 3D models which can be animated on the computer will either fall to pieces or be fixed in place when fabricated. Attaching the necessary fasteners and hinges via CSG operations involves tedious 3D manipulation and tends to result in sharp edges. Given a library of simple assembly connector surfaces , we can use drag-and-drop with rigid boundaries and faired transitions to quickly create assemblies suitable for fabrication (Fig. 7, 14).

Creative Compositions The simplicity of our interface provides a level of expressive freedom not available in previous composition tools. Some of the most positive feedback we received came from artists who wish to utilize 3D modeling in their physical works. The main effect of our tool seemed to be that it transformed tasks which had previously been tedious, error-prone, and best avoided if at all possible, turning them into enjoyable creative experiences. Fig. 15 shows a variety of creative models, some produced by the authors, and others by artists who have experimented with our tool.

4.1 Limitations

A fundamental limitation of our implementation is introduced by the use of the Discrete Exponential Map to compute local parameterizations. A fundamental property of the DEM, and our improved version, is that it will distort and even fold-over when passing across regions with widely varying curvature. This can cause visible tearing and even catastrophic failure in our local remeshing operations. A more robust local parameterization, or perhaps a scheme for local transport of the boundary across the surface, would be a significant improvement to our tool.

Two other similar problems were raised by professional artists, most of whom use rely almost exclusively on quad meshes for production models. We learned that these quad meshes contain carefully designed *edge loops* which support high-quality re-posing and animation, and that the areas where we introduce triangles to fill holes or stitch a part into the target mesh must be manually "re-topologized". Global quad-meshing algorithms do not appear to address these hole filling and stitching problems, as it is imperative that the existing quad mesh structure not be modified. Based on the feedback we collected, this appears to be the most pressing issue for artists and designers who have used our tool.

5 Discussion

We described two novel artist-oriented interfaces for shape composition: geometry drag-and-drop of arbitrary mesh parts, and a clone-brush for detail transfer. Based on Section 3 we implemented functional modeling tools and used them to demonstrate the utility of our methods in a variety of practical modeling workflows. These applications were gleaned from feedback provided by 3D professionals and hobbyists, whom we provided with a limited version of our drag-and-drop composition tool.

An interesting direction for future work is towards other simple tools to assist 3D modeling novices. Even the most intuitive sketch-based interfaces can provide little guidance in mastering the skills necessary to realize an idea as a detailed and realistic 3D model. In our experience, this steep learning curve is one of the most significant hinderances to wider artist adoption of 3D design. This barrier may perhaps be lowered if the novice is able to rely on libraries of parts to add the details they are not yet able to create themselves. We experienced this ourselves when creating Fig. 11.

Many 3D practitioners who tested our software expressed a strong desire for our techniques to be integrated into commercial tools. We close with an excerpt from detailed feedback provided by a modeling industry expert, who had this to say about our interface:

> [This] is one of the most interesting research software applications we have come across in many years. Ultimately creating good 3D models is actually an artistic and design process not a scientific and mathematical process. This is the only application that we have ever seen where we can just load up parts and add them together interactively. It gives the artist a freedom that has

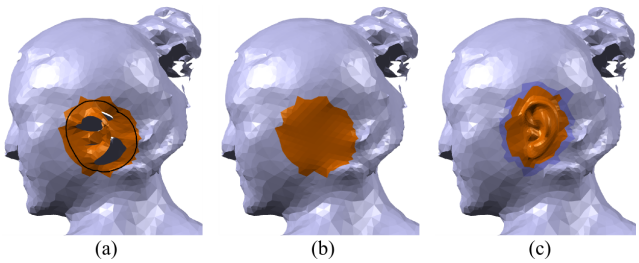

Figure 13: *Scanned data often includes areas with holes or missing detail (a). With our interface we can quickly drag off the invalid regions, leaving smooth fill surfaces behind (b). The missing detail can then be filled in using existing parts (c).*

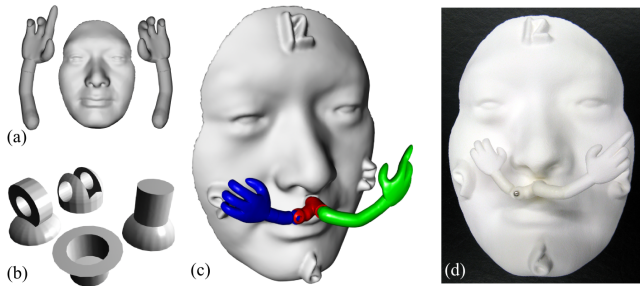

Figure 14: *Using a few parts (a) and a library of simple connectors (b), we were able to design a novel clock with a hinge in the hour hand, which allows time zone changes by bending the arm (c). We then fabricated a physical prototype with a 3D printer (d).*

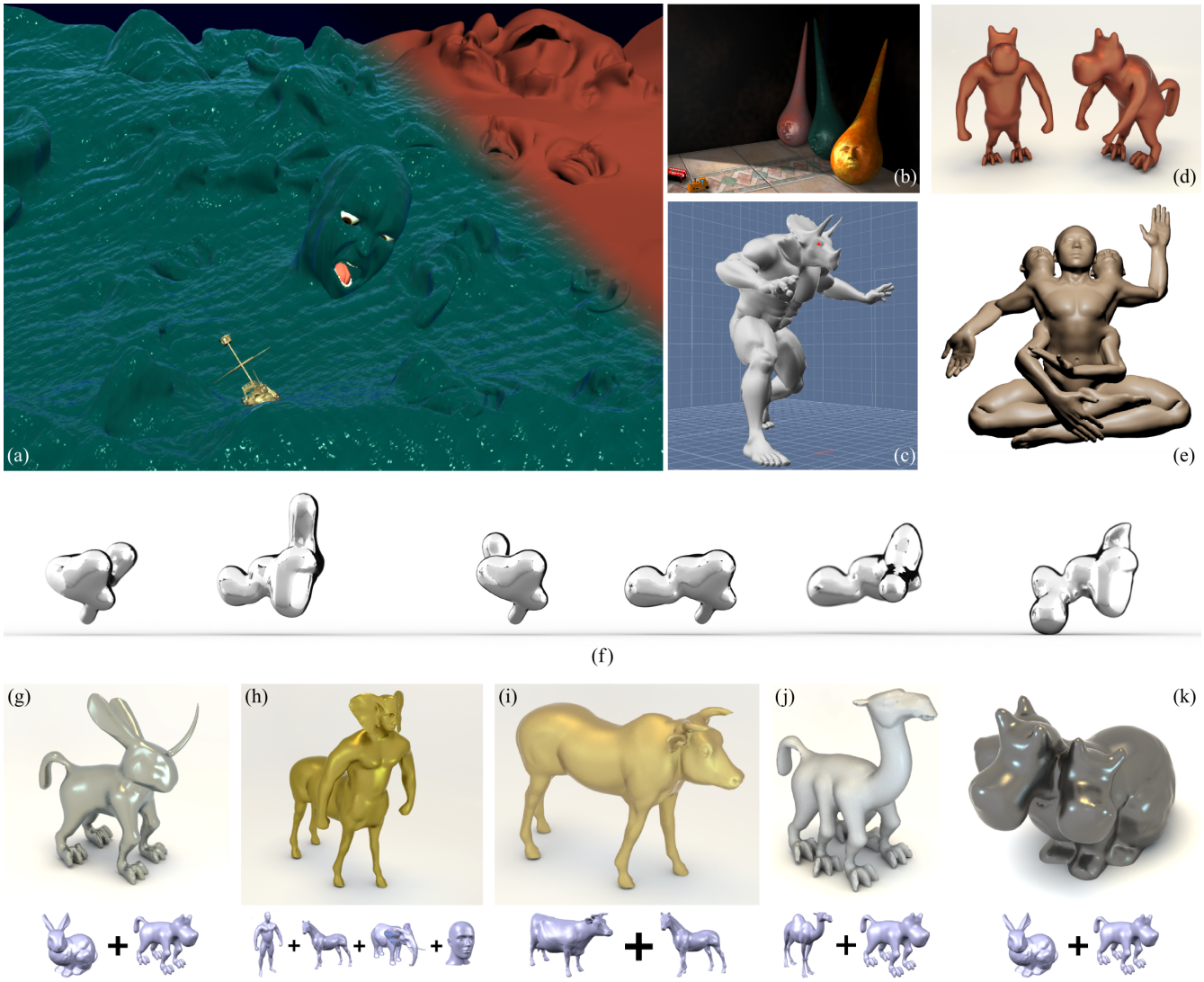

Figure 15: *Our drag-and-drop interface makes it simple and even enjoyable to create interesting 3D compositions.* "Sea of Faces" *(a) is composed of 42 parts and was created in under an hour by the second author, as was* "Bodhi" *(e).* "Toys for my Grandchild" *(b) and* "Tricerasapien" *(c) were contributed by testers of our software. A professional artist and jewelry designer sent us the results of a form exploration (f), using our tool to mix fluid simulation "spatters". The first author created a variety of mythical beasts, including Cynocephali (d), an Al-mi'raj (g), an Elephant-Eared Centaur (h), a Bonnacon (i), the elusive 6-legged Allocamelus (j), and the terrifying Cerebunny (k).*

not existed to date ... and returns a very enjoyable creative spontaneity.

References

ALEXA, M. 2002. Linear combination of transformations. *ACM Trans. Graph. 21*, 3, 380–387.

BARGHIEL, C., BARTELS, R., AND FORSEY, D. 1995. Pasting spline surfaces. In *Mathematical Methods for Curves and Surfaces*. 31–40.

BIERMANN, H., MARTIN, I., BERNARDINI, F., AND ZORIN, D. 2002. Cut-and-paste editing of multiresolution surfaces. *ACM Trans. Graph. 21*, 3, 312–321.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Trans. Vis. Comp. Graph. 14*, 1, 213–230.

BOTSCH, M., PAULY, M., WICKE, M., AND GROSS, M. 2007. Adaptive space deformations based on rigid cells. *Comp. Graph. Forum 26*, 3, 339–347.

DAVIS, J., MARSCHNER, S., GAR, M., AND LEVOY, M. 2002. Filling holes in complex surfaces using volumetric diffusion. In *Proc. Intl. Symp. 3D Data Proc., Vis., Trans.*

DESBRUN, M., MEYER, M., AND ALLIEZ, P. 2002. Intrinsic parameterizations of surface meshes. *Comp. Graph. Forum 21*, 209–218.

ELECTRONIC ARTS, 2009. Spore. http://www.spore.com.

FU, H., TAI, C.-L., AND ZHANG, H. 2004. Topology-free cut-and-paste editing over meshes. In *Proc. Geom. Model. and Proc. '04*, 173–182.

FU, H., AU, O. K.-C., AND TAI, C.-L. 2007. Effective derivation of similarity transformations for implicit Laplacian mesh editing. *Comp. Graph. Forum 26*, 1, 34–45.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph. 23*, 3, 652–663.

GAL, R., SORKINE, O., MITRA, N., AND COHEN-OR, D. 2009. iwires: an analyze-and-edit approach to shape manipulation. *ACM Trans. Graph. 28*, 3, 1–10.

GINGOLD, Y., AND ZORIN, D. 2008. Shading-based surface editing. *ACM Trans. Graph. 27*, 3, 1–9.

GINGOLD, Y., IGARASHI, T., AND ZORIN, D. 2009. Structured annotations for 2D-to-3D modeling. *ACM Trans. Graph. 28*, 5, 148.

GREENBERG, S., AND BUXTON, B. 2008. Usability evaluation considered harmful (some of the time). In *Proc. CHI '08*, 111–120.

HASSNER, T., ZELNIK-MANOR, L., LEIFMAN, G., AND BASRI, R. 2005. Minimal-cut model composition. In *Proc. SMI '05*, 72–81.

HORN, B. K. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer. 4*, 629–642.

HUANG, X., FU, H., AU, O. K.-C., AND TAI, C.-L. 2007. Optimal boundaries for Poisson mesh merging. In *Proc. SPM '07*, 35–40.

KANAI, T., SUZUKI, H., MITANI, J., AND KIMURA, F. 1999. Interactive mesh fusion based on local 3D metamorphosis. In *Proc. Graphics Interface '99*, 148–156.

KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. SIGGRAPH '98*, 105–114.

KRAEVOY, V., JULIUS, D., AND SHEFFER, A. 2007. Model composition from interchangeable components. *Proc. Pacific Graph. '07*, 129–138.

LIEPA, P. 2003. Filling holes in meshes. In *Proc. SGP '03*, 200–205.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. In *Proc. SIGGRAPH 2005*, 479–487.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2006. Laplacian mesh optimization. In *Proc. ACM GRAPHITE*, 381–389.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. FiberMesh: designing freeform surfaces with 3D curves. *ACM Trans. Graph. 26*, 3.

OSHER, S., AND FEDKIW, R. 2003. *Level Set Methods and Dynamic Implicit Surfaces*. Springer.

SCHMIDT, R., AND SINGH, K. 2008. Sketch-based procedural surface modeling and compositing using Surface Trees. *Comp. Graph. Forum 27*, 2, 321–330.

SCHMIDT, R., GRIMM, C., AND WYVILL, B. 2006. Interactive decal compositing with discrete exponential maps. *ACM Trans. Graph. 25*, 3, 605–613.

SHARF, A., ALEXA, M., AND COHEN-OR, D. 2004. Context-based surface completion. In *Proc. SIGGRAPH '04*, 878–887.

SHARF, A., BLUMENKRANTS, M., SHAMIR, A., AND COHEN-OR, D. 2006. SnapPaste: an interactive technique for easy mesh composition. *Vis. Comput. 22*, 9, 835–844.

SHEWCHUK, J. 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. vol. 1148 of *Lecture Notes in Computer Science*. Springer-Verlag, 203–222.

SINGH, K., AND FIUME, E. 1998. Wires: a geometric deformation technique. In *Proc. SIGGRAPH '98*, 405–414.

SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proc. Symp. Geom. Proc.*, 175–184.

SUMNER, R. W., SCHMID, J., AND PAULY, M. 2007. Embedded deformation for shape manipulation. *ACM Trans. Graph. 26*, 3.

SUZUKI, H., SAKURAI, Y., KANAI, T., AND KIMURA, F. 2000. Interactive mesh dragging with an adaptive remeshing technique. *The Visual Computer 16*, 3-4, 159–176.

YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2004. Mesh editing with Poisson-based gradient field manipulation. In *Proc. SIGGRAPH '04*, 644–651.

A Upwind-Average Discrete Exponential Map

The Discrete Exponential Map (DEM) approximates the exp/log map at a point $\mathbf{s}$, mapping the local neighbourhood into the *tangent space* $T_{\mathbf{s}}$ via propagation of *normal coordinates* outward from $\mathbf{s}$ [Schmidt et al. 2006]. Given a piecewise-linear path $\{\mathbf{s}, \mathbf{q}_1, \ldots, \mathbf{q}_n, \mathbf{p}\}$, computed via Dijkstra's algorithm, the tangent-space coordinates $T_{\mathbf{s}}\mathbf{p}$ are defined recursively as

$$T_{\mathbf{s}}\mathbf{p} = T_{\mathbf{s}}\mathbf{q} + M_{\mathbf{qs}} T_{\mathbf{q}}\mathbf{p} \tag{3}$$

where $T_{\mathbf{q}}\mathbf{p}$ is found via tangent-plane projection of $(\mathbf{p}\text{-}\mathbf{q})$, and $M_{\mathbf{qs}}$ is a 3D rotation aligning the tangent-normal frame $\mathcal{F}_q$ with $\mathcal{F}_s$.

One limitation of the DEM is that any error introduced at $\mathbf{q}_i$ will be propagated to downwind points whose path passes through $\mathbf{q}_i$, potentially leading to catastrophic failures (Fig. 16b). Similarly, as the paths are completely independent the error can vary wildly between two neighbouring points. Since the DEM sums vectors rather than scalars, $T_{\mathbf{s}}\mathbf{p}$ could be estimated from *any* nearby upwind point (Fig. 16). We re-define $T_{\mathbf{s}}\mathbf{p}$ as a weighted average:

$$T_{\mathbf{s}}\mathbf{p} = \sum_i w(\mathbf{p}, \mathbf{q}_i) \left(T_{\mathbf{s}}\mathbf{q}_i + M_{\mathbf{q}_i\mathbf{s}} T_{\mathbf{q}_i}\mathbf{p} \right) \tag{4}$$

where $\mathbf{q}_i$ are nearby upwind neighbours to $\mathbf{p}$ (Fig. 16b) and w is the inverse distance weight As shown in Fig. 16, upwind averaging improves DEM robustness, with a small 5-10% increase in runtime cost.

As the DEM uses normal information, smoothing the normal field relaxes the parameterization in regions of higher curvature. Replacing each normal with a distance-weighted average of k-neighbourhood can be efficiently evaluated in-line with the DEM, and combind with upwind averaging, results in much more stable maps (Fig. 16).

B COILS Deformer

The COILS deformer is based on a rotation-invariant formulation of the well-known WIRES deformer [Singh and Fiume 1998]. Assume we have a piecewise-linear curve C, and each vertex $\mathbf{q}$ of the

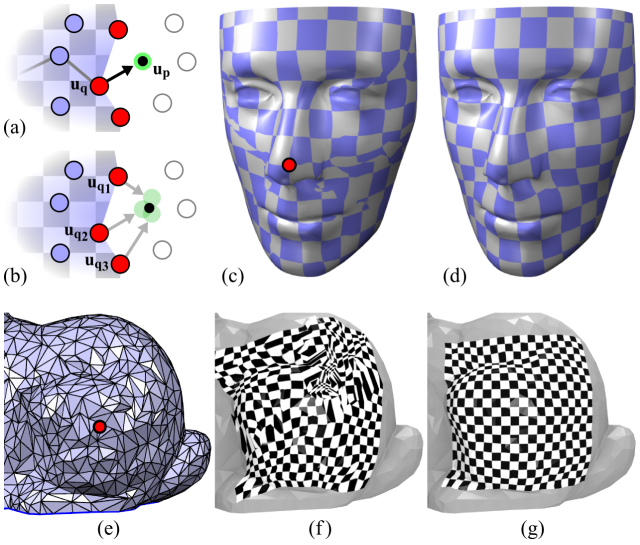

Figure 16: *In our improved Discrete Exponential Map, instead of estimating the uv-parameter $\mathbf{u}_p$ from a single upwind sample $\mathbf{u}_q$ (a), we average multiple estimates (b). Compared to the original DEM (c,f), this simple modification is much more robust (d,g).*

curve has an arbitrary orthonormal coordinate frame $\mathcal{F}_\mathbf{q}$ (ie a 3D rotation matrix with the frame vectors as rows). Then any other point $\mathbf{p}$ can be expressed as $\mathbf{q}+\mathcal{F}_\mathbf{q}\mathbf{v}(\mathbf{p},\mathbf{q})$, where $\mathbf{v}(\mathbf{p},\mathbf{q}) = \mathcal{F}_\mathbf{q}^{-1}(\mathbf{p}-\mathbf{q})$ is a vector in the frame $\mathcal{F}_\mathbf{q}$. Given deformed positions $\widehat{\mathbf{q}}_i$ and frames $\mathcal{F}_{\widehat{\mathbf{q}}_i}$, the deformed position $\widehat{\mathbf{p}}_\mathbf{q}$ is:

$$\widehat{\mathbf{p}} = \sum_\Omega \frac{w(\mathbf{p},\mathbf{q}_i)}{\sum_\Omega w(\mathbf{p},\mathbf{q}_i)}(\widehat{\mathbf{q}}_i + \mathcal{F}_{\widehat{\mathbf{q}}_i}\mathbf{v}(\mathbf{p},\mathbf{q}_i)) \tag{5}$$

where Ω is the vertices of C, w is the inverse squared distance weight $(d(\mathbf{p},\mathbf{q})^k + \epsilon)^{-1}$, and d is either Euclidean or geodesic distance.

If C is the open boundary $\partial\mathcal{S}$ of a mesh patch $\mathcal{S}$, we can deform the interior using the above technique, however significant loss of volume occurs with large deformations (Fig. 17g). Re-factoring Equation 5 into the sum of a weighted centroid and an average displacement vector, we note that as points increase in distance from the boundary, weight is more evenly distributed over $\partial\mathcal{S}$, pulling the centroid down and causing vertical squashing.

For points close to the boundary, weight is concentrated nearby and the centroid is relatively static. Hence, we should deform each interior point relative to a nearby surface region $\Omega(\mathbf{p})$. Conceptually, we compute a front propagating away from the boundary, and deform points on the front at timestep t relative to the *upwind* front at $t-1$. Since our surface is irregularly sampled, we define t at vertices using approximate geodesic distance $g_{\mathbf{p}_i}$ to the boundary, computed using Dijkstra's algorithm, and then represent the front by a band of upwind points. We define a geodesic band of width r_u, taken to be a small multiple (we use 2.1) of the average edge length. Then $\Omega(\mathbf{p}_i) = \{\mathbf{p}_j : g_{\mathbf{p}_i} - r_u < g_{\mathbf{p}_j} < g_{\mathbf{p}_i}\}$.

To reduce the effects of sampling variation, we modulate our weighting function, adapting the "smeared-out" Heaviside functions used in level set front propagation [Osher and Fedkiw 2003]. Given a function which smoothly falls off from 1 to 0, such as $f(x,r) = \max((1-x^2/r^2)^3, 0)$, we define the upwind weight:

$$\triangle_g = g_{\mathbf{p}_i} - g_{\mathbf{p}_j} \qquad r_n = \min_{i\neq j}|\mathbf{p}_i - \mathbf{p}_j| \tag{6}$$

$$w_A(\mathbf{p}_i,\mathbf{p}_j) = f(\triangle_g, r_u)\,(1 - f(\triangle_g, r_n)) \tag{7}$$

where the first term falls off away from the front and the second reduces the weight on points whose arrival time is nearly the same as at $\mathbf{p}_i$, mitigating potential bias effects. To correct for sampling variation the weight is modulated with a regularization factor $w_R(\mathbf{p}_i) = \min_{\mathbf{p}_k\in\Omega(\mathbf{p}_i)}|\mathbf{p}_k - \mathbf{p}_i|^2$. Our final weighting scheme is then $w_C = w_A \cdot w_R \cdot w$.

Note that $\mathbf{p}_i$ is now deformed relative to other internal points, but transformed frames are only provided at boundary samples. Hence, we also apply Equation 5 to blend relative frames: $\mathcal{F}_{\widehat{\mathbf{p}}} = \sum_{\mathbf{q}_i\in\Omega(\mathbf{p})} w_i \mathcal{F}_{\widehat{\mathbf{q}}_i}\mathcal{F}_{\mathbf{q}_i}^{-1}\mathcal{F}_\mathbf{p}$. Qualitatively, logexp blending [Alexa 2002] was only slightly stiffer than a much faster orthonormalized linear blend, so we use the latter.

This method is very stable, visually approximates the results of least-squares differential techniques, and can be applied to arbitrary kNN graphs (useful for non-manifold or disconnected meshes, or even point sets). The main limitation is that discontinuities in the geodesic field are reflected in the deformation, so the deformation is non-smooth. In particular, where the geodesic field merges after having split to pass around a hole or handle, catastrophic discontinuities can occur. Including disconnected components in the upwind band avoids the latter problem, at the cost of allowing influence from semantically-disconnected components.

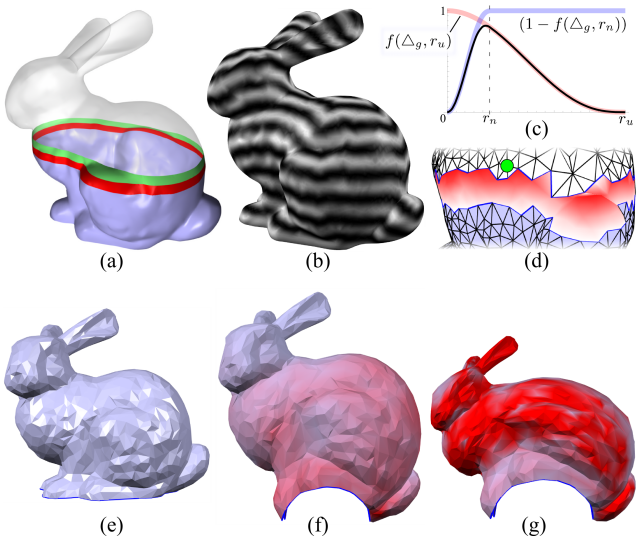

Figure 17: *Conceptually, the COILS deformer slices the bunny into layers (a) and deforms layer t (green) relative to layer t-1 (red). On a mesh we combine the geodesic distance to the boundary (b) with a "smearing" function (c) to approximate layer t-1 at the green point (d). For even moderate deformations (e) the COILS deformation (f) is much more robust than the WIRES deformer (g).*