Mesh Composition on Models with Arbitrary Boundary Topology

Juncong Lin, Xiaogang Jin, Charlie C.L. Wang*, and Kin-Chuen Hui

Abstract—This paper presents a new approach for the mesh composition on models with arbitrary boundary topology. After cutting the needed parts from existing mesh models and putting them into the right pose, an implicit surface is adopted to smoothly interpolate the boundaries of models under composition. An interface is developed to control the shape of the implicit transient surface by using sketches to specify the expected silhouettes. After that, a localized Marching Cubes algorithm is investigated to tessellate the implicit transient surface so that the mesh surface of composed model is generated. Different from existing approaches in which the models under composition are required to have pairwise merging boundaries, the framework developed based on our techniques have the new function to fuse models with arbitrary boundary topology.

Index Terms—Mesh composition; Arbitrary boundary topology; Modeling by silhouette; Localized Marching Cubes; Implicit surface.

1 Introduction

RESEARCHES for providing a user-friendly modeling system to create complex 3D models effectively have been studied with a long history in computer graphics. It is common for designers to start modeling with a vague image of the shape in mind with reference to existing models or the features of models. The geometric modeling interface is expected to have the ability to capture design features of shapes and form a new shape that inherits the features of existing shapes. The term *design feature* here does not refer to the features such as holes or drains but to the aesthetic elements with geometry details. The widely investigated mesh composition approach is for this purpose. However, current existing approaches (ref. [3], [12], [14], [17], [24], [45], [52]) are with constraints at the topology of boundaries on the models under composition. More specifically, pairwise merging boundaries are usually expected. Although the approaches using implicit surfaces (such as [22], [29], [30] and [44]) can somewhat solve the problem, the models constructed by them do either need to have simple transient surface shape (as in [22] and [44]) or convert all meshes into implicit representation (ref. [29] and [30]) which easily yields shape approximation errors on the parts under composition. The work presented in this paper aims at overcoming above drawbacks and providing a more powerful modeling interface for creating complex 3D models from the existing ones.

For example as illustrated in Fig.1, the composed models by our approach are with arbitrary boundary topology. After cutting the needed parts from existing mesh models and putting them into the right pose, the Radial Basis Function (RBF) based implicit surface [48] will be employed as the transient surface to interpolate the boundaries of models under composition smoothly. Simply using the RBF-based implicit surface will have no shape control on the transient surface. To solve this problem, a new interface is developed to control the shape of transient surface by using sketches to specify the expected silhouette. The most difficult part of using implicit surface to fuse mesh models comes from how to convert the implicit surface into a mesh consistent with the boundary of given models under composition. A localized Marching Cubes algorithm has been developed in this paper for this purpose. Lastly, a method based on Laplacian mesh processing [45] has been developed as a post-processing step to reconstruct geometry details on the tessellated transient surface.

1.1 Previous Work

The work proposed in this paper relates to several previous researches in the areas of: modeling by examples, model completion and repair, similarity-based mesh editing and implict surface tessellation.

Modeling by example

Several approaches in literature (ref. [3], [12], [14], [17], [22], [24], [29], [30], [43], [45], [52]) explore a design by example strategy to construct complex 3D models from existing ones. The methods presented in [3], [12] and [24] focused on the cut-and-paste operation on meshes. They could produce seamless composed models. However, the joined models in their approaches were required to have boundaries topologically equivalent to a disk for the necessary mapping between source and target models. The methods in [45] and [52] can transplant and merge meshes also. However, these methods were actually deformation based; thus, boundary openings with similar topological structures were required. In more detail, it is hard to directly join two models where one has two openings while the other is only with one opening. Funkhouser et al. in [14]

*Corresponding Author.

Juncong Lin and Xiaogang Jin are currently with the State Key Lab of CAD&CG, Zhejiang University, P.R.China; Charlie C.L. Wang and Kin-Chuen Hui are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, P.R.China. Part of the work is finished when Juncong Lin was a research assistant in The Chinese University of Hong Kong.

Manuscript received (insert date of submission if desired). Please note that all acknowledgments should be placed at the end of the paper, before the bibliography.

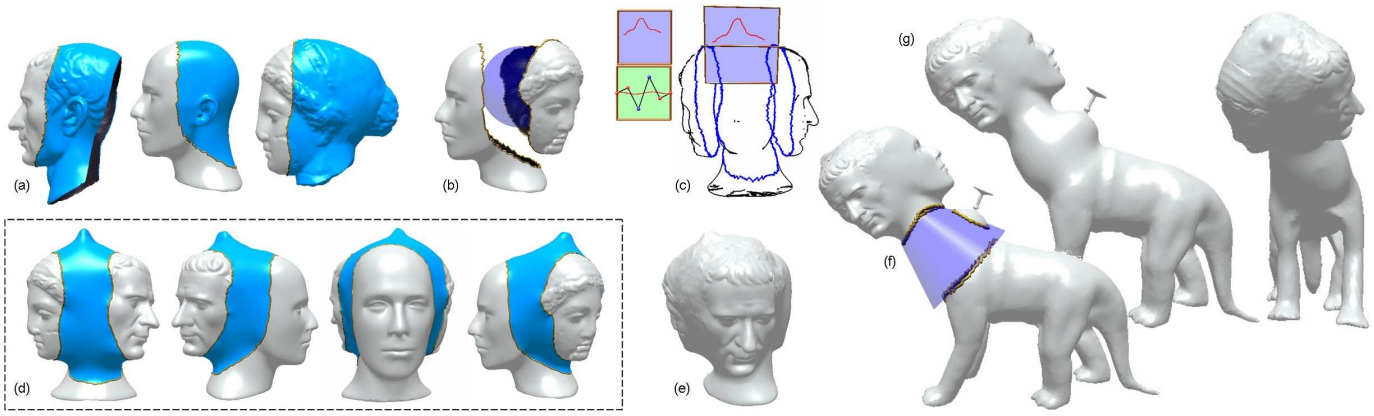

Fig. 1. An example to illustrate the functionality of our mesh composition framework: (a) the faces are scissor from three existing head models; (b) the faces under composition are placed together around a sphere and the position of faces can be easily changed on the sphere; (c) after specifying the silhouette of transient surface – so that form a horn on the top of the three-head model, the mesh of transient surface is generated by our localized Marching Cubes algorithm in (d); (e) we can then further propagate geometry details onto the transient mesh surface; (f) we then place the three-head model on to the neck of a dinosaur around a truncated cone together with some other decoration – again, the models can be easily moved according to the truncated cone; (g) the final three-head monster model can be generated effectively.

investigated a data-driven approach to construct new 3D models by assembling parts from existing ones. Recently, Hassner et al. [17] applied the graph minimal-cut operation to the task of 3D model composition. Both [14] and [17] used the technique in [24] to stitch those retained portions together. In summary, there are various topological limitations in all above methods. Recently, Sharf et al. [43] devoted to reduce the burden of users in placing models. Stimulated by the snapping notion which has been vastly used in graphics applications, they extended the notion to 3D to solve both the positioning and blending problem of two surfaces by a soft Iterative Cloesest Point (ICP) algorithm. Their tool allowed even little kids to construct complex models easily. Another recent work to automatically adjust the shape of merging boundary is presented in [19]. However, they are still strict with the topology of boundaries. The topology constraints embedded in above approaches can be somewhat overcome by the implicit surface based approaches (e.g., [22], [29], [30] and [44]). However, the models constructed by [22] need to have planar openings which limit the type of models that can be created. The approach in [44] limited the transient implicit surface to be polyhedral primitives, whose shape is too simple. The authors in [29] and [30] convert models into implicit representation and merge them. Although mesh surface can be reconstructed from the implicit representation, this explicit-implicit-explicit conversion introduces unexpected shape approximation errors. All these limitations will be overcome by our scheme in this paper.

Model completion and repair

Model completion usually has two requirements: 1) boundary based – the completed surface patch should seamlessly match the boundary, and 2) context based – the patch should also contain geometry details similar to those on the existing surfaces. Earlier researches all focused on the boundary condition. The authors of [7] and [49] conducted the completion by solving certain partial differential equations, while others (e.g., [23] and [38]) were volumetric methods that represented the surface as the boundary between inside and outside volumetric regions. Recently, several context-based methods were proposed in [26], [32], [35] and [42] to repair holes according to surface context information. Kraevoy and Sheffer [26] filled the holes using a mapping between the incomplete mesh and a template model. Sharf et al. in [42] extended texture synthesis techniques from 2D to 3D, and represented intrinsic geometry properties and performed geometry completion directly in the 3D domain. In [32] and [35], the 3D geometry synthesis problem is transformed into a 2D domain using a conformal parameterization algorithm and then solved in the target regions using image completion techniques. These approaches all focus on the completion of holes on one model. Differently, our approach will solve the problem of geometry completion among several models with topological incompatible merging boundaries, and use sketches like the ones in [20] and [25] to control the shape of transient surface.

The mesh composition method presented in this paper is also similar to the model rapir approaches using volumetric techniques (e.g., [5] [23] [34] [56]), where volumetric representation is conducted to fill holes and fix topological errors on a given model. Here, we employ implicit surfaces to interpolate the gap between composing models with arbitrary boundary topology.

Similarity-based mesh editing

Recently, semi-local similarity based shape descriptor has been investigated in the applications of shape matching, retrieval, modeling and smoothing (e.g., [16], [42], [51], [53] and [54]). Zelinka and Garland [53] proposed a concept of *geodesic fans* to faithfully identify regions on a surface that are geometrically similar. Their descriptor is a vector of n discrete samples taken at n fixed sample positions according to some sampling pattern given in geodesic polar coordinates. Based on [53], they later presented a curvature map method in [16] to create the unique signature for a surface point. They also adopted a similar technique to transfer geometry details onto models in [54]. The post-processing geometry detail reconstruction step in our mesh composition framework borrows some idea from [16], [53] and [54] but using a different shape descriptor. Recently, a similarity based mesh smooth-

ing method has been presented in [51], which extends the NL-means image filter to the piecewise smooth surface represented by triangle soups.

Implicit surface tessellation

Marching cubes (MC) algorithm was first introduced by Lorensen and Cline [28] and has become the most commonly used method for tessellating implicit surfaces. As first noted by Duerst [10], the original MC algorithm [28] may produce isosurfaces with holes due to topologically inconsistent decisions on the reconstruction of ambiguous faces, where the borders used by one incident cube do not match the borders of the other incident cube. Several approaches addressing this problem have been published (see [2] and [33] for a review). Recently, Lewiner et al. [27] presented an efficient and robust implementation of Chernyaev's Marching Cubes 33 algorithm [6] which ensures a topologically correct tessellation. None of above MC algorithm and it variants cover the problem of tessellating a part of implicit surface. In this paper, we develop a new algorithm based on the configuration table in [27] to tessellate the portion of interest on the transient implicit surface. Although the authors in [36] and [40] also presented method to trim an implicit surface by another implicit, our algorithm presented in this paper is the first localized Marching Cubes algorithm to generate meshes of a partial implicit surface trimmed by an explicit mesh surface.

Marching triangles (MT) is another class of isosurface approximation approaches. It is firstly appeared in [18]. The MT algorithm employs the local 3D constraint to reconstruct a Delaunay triangulation of an arbitrary topology manifold surface. This method is further enhanced in [1] and [41] by adapting the size of triangles to the curvature of surface and closing cracks at the end of mesh growing. However, the drawback inherent to all continuation methods still exists that it is difficult to determine seed triangles on the connex part of a surface. Furthermore, the MT-like algorithms depending on seed searching are usually much slower than the MC algorithm and its variants.

1.2 Contributions

The approach presented in this paper has the following technical contributions.

- *Silhouette-based shape modeling of transient implicit surface*: We develop a method to control the shape of RBF-based implicit transient surface, where the shape of transient surface can be easily controlled by using sketches to specify the expected silhouettes.
- *Localized Marching Cubes Algorithm*: In order to tessellate the transient implicit surface into a mesh connecting the models under composition, a localized Marching Cubes algorithm is investigated. Although it is not an absolutely new algorithm, this localization is the first method to contour trimmed implicit surface with complex boundaries.

Based on these two technical contributions, a new model composition framework can be constructed to fuse models with arbitrary boundary topology, which is a function that has not been provided in existing model composition approaches in literature.

Rest of the paper is organized as follows. Section 2 presents the RBF implicit surface based shape modeling of transient surface. The localized Marching Cubes algorithm is then detailed in section 3. Section 4 describes other algorithms for the mesh composition framework. Experimental results and necessary discussions are given in section 5. Lastly, our paper ends with the conclusion section.

2 Shape Modeling of Transient Surface

We begin this section with the reasons for selecting RBF-based implicit surface and a brief of its mathematical formulas, and then describe how it is applied to model the transient surface interpolating the merging boundaries on given models and how the shape control of silhouettes can be given.

2.1 RBF-based Implicit Surface

There are many different implicit surfaces in literature. By investigation, we find that *Radial Basis Functions* (RBF) based implicit surface is the best candidate for mesh composition on models with multiple boundary openings since it holds the following two important strengths.

- Both the position and the normal at a surface point can be specified on a RBF-based implicit surface so that we can generate a transient surface smoothly interpolating the boundary of models under composition – specified normal vectors are conducted to ensure the tangential smoothness. Also, because of the ability to control both positions and normals, it provides us the possibility to control the shape of surface silhouettes.
- A RBF-based implicit surface can describe a closed two-manifold surface with arbitrary topology in a unique and compact mathematical representation – so that the variation of topology can be easily implemented on this mathematical representation, which leads to a new function that is not provided on other existing mesh composition paradigms.

Following [48], the RBF-based implicit surface is based on the thin-plate interpolation and can be expressed by a weighted sum of appropriate radial basis functions plus an affine term as

$$\Gamma(v) = p(v) + \sum_{i=1}^{N} \lambda_i \phi(\|v - v_i\|), \quad (1)$$

where λ_i s are weights, $\{v_i = \{x_i, y_i, z_i\}\}_{i=1}^{N}$ are the location constraints, and $p(v)$ is a linear affine function of position in the form of $p(v) = p_0 + p_1 x + p_2 y + p_3 z$. For the radial basis functions, we employ $\phi_i(r) = r^3$ which is widely used in 3D problems. For N position constraints, they can be written as N linear equations

$$\Gamma(v_j) = p(v_j) + \sum_{i=1}^{N} \lambda_i \phi(\| v_j - v_i \|) = f_j \quad (2)$$

with $j = 1,...,N$ and f_j is the function value shown at the location v_j. The unknown λ_i s in above linear equation system can be solved by adding the following compatible condition

$$\sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \lambda_i x_i = \sum_{i=1}^{N} \lambda_i y_i = \sum_{i=1}^{N} \lambda_i z_i = 0 .$$

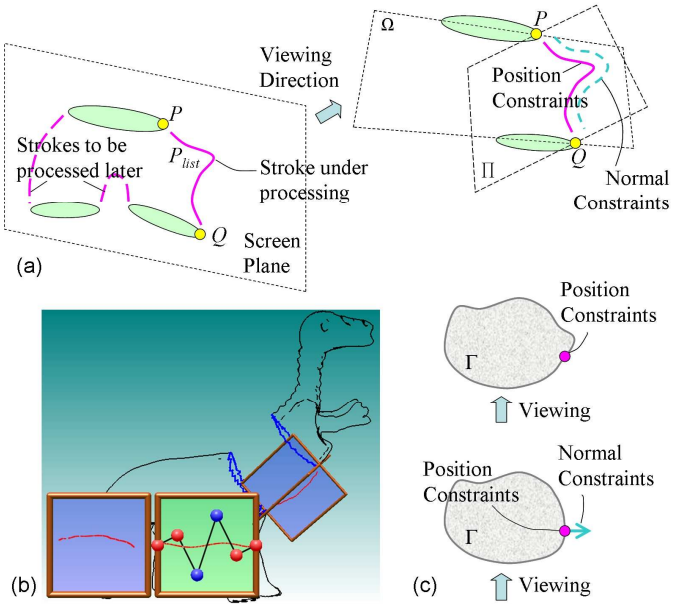

Fig.2. Illustration for determining the 3D shape of silhouette curve through sketching: (a) the method to determine depth coordinates, where two planes – the *projection plane* and the *depth plane* are conducted; (b) the projection of silhouette curve on the projection plane (in blue) and the depth plane (in green) can be smoothed with the help of spline curve fitting and adjusted with the control points; (c) cross-section views of Γ by simplify asking the transient surface to only interpolate sampled position constraints (top) versus the transient surface with normal constraints added (bottom) – it is clear that we cannot keep the sampled points on surface silhouette without normal constraints.

2.2 Shape Modeling

Equipped by the RBF-based implicit surface modelling method, the transient surface Γ to join the scissored models can be elegantly constructed. The shape of transient implicit surface is controlled by two types of constraints: the boundary constraints and the silhouette constraints.

To achieve position interpolation on the boundaries of models under composition, all vertices on the boundaries are substituted into Eq.(2) where each yields a linear equations. Since the transient surface should exactly pass all the vertices on merging boundaries, the function values of Γ shown on the vertices are zero (i.e., $f_i = 0$). However, simply setting these position constraints leads to vanish side conditions so that the computation of Γ fails. Based on this reason and also to ensure the smooth interpolation, Turk and O'Brien in [48] added the normal constraints to the linear equation system. Here, we also define normal constraints in a similar manner but with more points. Specifically, we first add all vertices of the merging boundary and their n-rings neighbors on the models under composition into the RBF system as interpolating points. $n = 3$ is chosen since it achieves a balance of the computational speed and the smoothness of reconstructed implicit surface. The set of all these vertices are denoted by V^*. Then, for each vertex $v_i \in V^*$ and its surface normal n_i, a normal constraint is added by placing a position constraint at $v_i + \tau n_i$ with the function value $f_i = \tau$ where τ is a small value. We use $\tau = 0.1 L_{\min}$ in all of our examples, where $L_{\min}$ is the shortest (non-zero) edge length on boundaries. The surface normal at a vertex is computed by angle-weighted average of the normals [21] on its adjacent faces. Using large value of τ may make the normals intersect each other, therefore a small value is chosen for τ. Above normal constraints are added to ensure the smooth transition in mesh composition. In some cases, if nonsmooth transition is wanted, we can adjust the orientation of n_i to satisfy different interpolation requirements.

In order to control the shape of transient surface Γ, the silhouette constraints are added into the RBF system to vary the shape of Γ only determined by boundary constraints. A silhouette curve of Γ is determined by two steps through sketching. After drawing a stroke on the screen plane – the stroke is stored as a list of points P_{list}, we firstly find the two merging boundaries A and B by the closest boundary points to the starting and ending points of P_{list} in the screen plane. Note that A and B could be on the same model under composition. The two ending points P and Q on the silhouette of these two openings are then located. After that, a plane Ω containing P and Q, and whose normal vector is closest to the viewing vector, is determined – the plane is called as *projection plane*. All points in P_{list} are firstly projected onto Ω to get temporary depth coordinates. Secondly, we construct another plane Π, named as *depth plane*, passing P, Q and the viewing vector. The reason for introducing Π is to give users an interface to control the shape of silhouette better in 3D. The points in P_{list} on Ω are then further projected onto Π to determine their 3D coordinates. A sixth-order Bezier curve C_f with 6 control points are employed to approximate the projected points in P_{list}. Among the six control points, two control points at each end of the curve (red ones in Fig.2(b)) are fixed to ensure the tangential smoothness. The left two control points (blue ones in Fig.2(b)) are allowed to move to further modify the shape of silhouette curve. The coordinate of control points can be determined and adjusted by their projection on Ω and Π. Points are uniformly sampled on C_f to serve as position constraints of the silhouette. Simply asking the transient implicit surface Γ to interpolate the sampled points from C_f cannot ensure that they are on the silhouette of Γ (see Fig.2(c)). From the definition of silhouette in computer vision, we know that for a point on a silhouette, the surface normal at this point is perpendicular to the viewing direction. Therefore, the normal constraints on the sample points from C_f also need to be added. With the tangent vector t_p at a sampled point $p \in C_f$ and the viewing vector n_v, the normal constraint at point p (i.e., $n_p = t_p \times n_v$) is added to model the RBF-based implicit surface Γ. It is easy to conduct silhouettes to control the shape of transient implicit surface. Examples on the shape modeling of transient implicit surface will be shown later in the experimental result section.

Position and normal constraints on both the surface boundaries and the silhouette curves usually result in about several thousand linear equations in the RBF system. To efficiently solve this dense linear equations system, the public available library LU-GPU [15], which is accelerated by GPUs, is employed. A dense linear equation system with dimension 3,000 can be solved in less than 5 seconds (i.e., in an interactive speed).

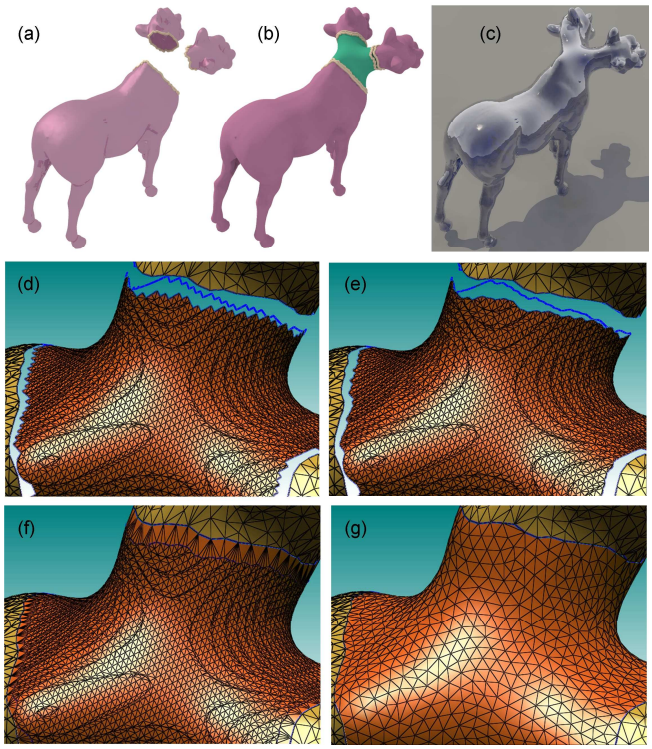

Fig.3. Tessellating the transient implicit surface using our new Localized Marching Cubes algorithm: (a) the models with nonpairwise boundaries under composition, (b) partially finished transient mesh surface – gap strips are left to be further triangulated, (c) the finished model, (d) the zoom and mesh view of (b) – the blue curve is the back of the boundary on the other side, (e) the zigzag boundary in (d) are smoothed, (f) progressive result with gap strips triangulated, and (g) the final remeshed transient surface.

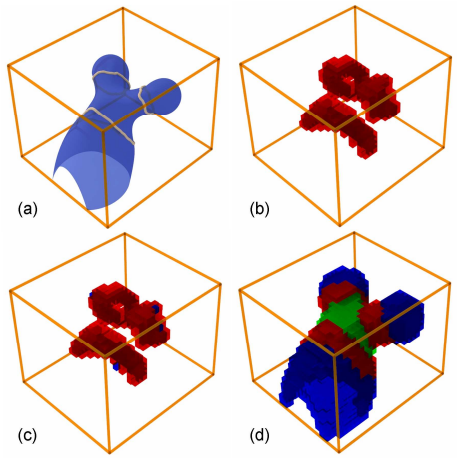

Fig.4. Illustration for the cubes construction and classification in our *Localized Marching Cubes* algorithm: (a) the transient implicit surface in the working space are separated into portions by the boundaries, (b) the boundary surface cubes – *BS-Cubes*, (c) some seed *IS-Cubes* (in blue) are found near *BS-Cubes*, and (d) the *VS-Cubes* (in green) and the *IS-Cubes* (in blue) are separated through a flooding algorithm with the seed *IS-Cubes*.

3 Localized Marching Cubes Algorithm

The transient surface constructed so far is represented implicitly in a scalar function. We need to tessellate the transient surface into meshes. However, as the transient surface Γ is a closed implicit surface partitioned by the boundaries on models under composition, only part of the implicit surface belongs to the transient region. Therefore, a tessellation method is needed to provide the following functions:

- Distinguish the interested and non-interested regions on the transient implicit surface automatically;
- Construct mesh compatible to the mesh connectivity on boundaries of models under composition;
- Generate well-shaped triangles.

Existing implicit tessellation methods rarely satisfy all these requirements. We propose a new algorithm – the *Localized Marching Cubes* algorithm plus remeshing for solving the above issues. Our algorithm consists of four steps: 1) Cubes construction and classification, 2) Topology guaranteed tessellation (an example of the step result has been illustrated in Fig.3(b) and 3(d)), 3) Quality optimized gap triangulation (an example step result is shown in Fig.3(f)), and 4) Transient surface remeshing (resulting in Fig.3(g)).

3.1 Cubes Construction and Classification

The algorithm proposed here is a variation of the famous Marching Cubes (MC) algorithm, whose spirit is to subdivide the interested space Ψ of an implicit surface into cubic sub-spaces (named as cubes) and then tessellate the surface based on the *inside*/*outside* flags of the eight nodes on each cube. Similarly, the first step of localized MC is also to construct cubes. However, different from original MC, we need further classify the constructed cubes into different categories so that can trim the transient implicit surface Γ into interested and non-interested portions. In detail, four types of cubes are defined in our algorithm:

- *E-Cube*: For a cube, if its eight nodes are all *inside* Γ (i.e., $\Gamma(p)>0$) or all *outside* Γ (i.e., $\Gamma(p)<0$), the cube will be empty during the tessellation – denoted by *E-Cube* as no triangle will be generated in these cubes. All the other cubes are surface cubes, *S-Cube*, which can be further classified into three types below.
- *BS-Cube*: For an *S-Cube*, if it intersects any boundary edge on the models under composition, it is defined as a *boundary intersecting cube*. For any *S-Cube* not intersecting the boundaries, if one of its 26 neighboring cubes is a *boundary intersecting cube*, this *S-Cube* is defined as a *boundary neighboring cube*. To be robust in the later region separation, both the *boundary intersecting cubes* and the *boundary neighboring cubes* are classified into boundary surface cubes, *BS-Cube*. The red cubes in Fig.4 are *BS-Cubes*.
- *VS-Cube* and *IS-Cube*: With the *BS-Cubes*, the left *S-Cubes* are then separated into the valid surface cubes (named as *VS-Cube*) that lie on the interested region of the transient implicit surface Γ and the invalid surface cubes (called *IS-Cube*) on the non-interested region of Γ. *VS-Cubes* are shown in green color in Fig.4 while *IS-Cubes* are displayed in blue.

Note that in the mesh composition framework, the working space Ψ of our localized MC is a region slightly larger than the bounding box of all merging boundaries on the models, and the size of cubes are selected as 0.8 times of the average length of edges on the merging boundaries. The cubes in Fig.4 are shown with a larger size for illustration.

We conduct a flooding algorithm to separate the *IS-Cubes* from the *VS-Cubes*. Firstly, the *BS-Cubes* are clustered into groups. For example there are three groups for the *BS-Cubes* shown in Fig.4(b). On the models under composition, 3-rings of triangles near to the merging boundaries are named as *boundary triangles*. Any type-undetermined *S-Cube* intersecting a boundary triangle can serve as a seed *IS-Cube*. Starting from the seed *IS-Cube*, a flooding algorithm can be used to find all other *IS-Cubes* linked to this seed *IS-Cube*. The steps of finding seed *IS-Cubes* and flooding are repeated until no more seed *IS-Cubes* can be found. That means we have determined all *IS-Cubes* in the working envelope – e.g., the blue cubes in Fig.4(d). The blue cubes in Fig.4(c) illustrate the seed *IS-Cubes* adopted for flooding. Then, all the left type-undetermined *S-Cubes* are classified into *VS-Cubes* (e.g., the green ones in Fig.4(d)). The pseudo-code of the seed *IS-Cube* search algorithm – **Function** *SeedCubeSearch* (*c*) and the *IS-Cube* flooding algorithm – **Function** *CubeFlooding* (*s*) is listed below. Both are implemented as recursive functions.

Function *SeedCubeSearch* (*c*)
Input: a *BS-Cube c*
Output: a seed *IS-Cube*
1. **for** any of *c*'s 26 neighbors – c_n
2. **if** c_n is *E-Cube* OR *BS-Cube* OR *IS-Cube*, **then continue**;
3. **if** c_n intersect a boundary triangle, **then return** c_n;
4. **for** any of *c*'s 26 neighbors – c_n
5. *s* = *null*;
6. **if** c_n is a *BS-Cube*, **then** *s* = *SeedCubeSearch* (c_n);
7. **if** *s* is NOT *null*, **then return** *s*;
8. **return** *null*;

Function *CubeFlooding* (*d*)
Input: a seed *IS-Cube d*
Output: the flooding result
1. Assign *d* to an *IS-Cube*;
2. **for** any of *d*'s 26 neighbors – d_n
3. **if** d_n is *E-Cube* OR *BS-Cube* OR *IS-Cube*, **then continue**;
4. *CubeFlooding* (d_n);

For each group of *BS-Cubes*, we just randomly select one *BS-Cube c* as input to call **Function** *SeedCubeSearch* (*c*). A seed *IS-Cube s* will be determined by this function. We then call **Function** *CubeFlooding* (*s*) to propagate the region of *IS-Cubes*. If the return of **Function** *SeedCubeSearch* (*c*) is *null*, we need to select a *BS-Cube* from another group to find the seed *IS-Cube* by calling this function again. The seed cube search and the cube flooding will be iteratively conducted until no more seed *IS-Cube* can be found. In both of these two functions, the searches are conducted locally. Thus, the classification procedure can be completed quickly.

3.2 Topology Guaranteed Tessellation of *VS-Cubes*

In this step, triangles are generated in the *VS-Cubes* based on the efficient and robust implementation [27] of the Marching Cubes 33 algorithm in [6]. The algorithm runs through 4 steps for each *VS-Cube*:

- *Step 1*) The case number and configuration is determined based on the *inside*/*outside* flags on eight nodes of the *VS-Cube*;
- *Step 2*) Looking up faces that need to be further tested in the above determined configuration;
- *Step 3*) For each face needs to be further tested, testing and determining its corresponding subcase;
- *Step 4*) Looking up the tiling for a determined subcase and generating triangles in this *VS-Cube*.

This implementation in fact depends on three tables:

- A *case table* maps each of the 256 possible configurations of a cube to one of the 15 cases and to a specific number designating this configuration;
- A *test table* stores the further tests to be performed to resolve topological ambiguity for each configuration;
- A *tiling table* encodes the method to triangulate a cube for each configuration and subcase.

The tessellation results are guaranteed to be a two-manifold mesh surface. As only *VS-Cubes* are tessellated, there is a gap between the given model and the resultant mesh (e.g., see Fig.3(d)), where is occupied by *BS-Cubes* and will be further triangulated in the following step.

3.3 Quality Optimized Gap Triangulation

The resultant mesh surface from previous step is usually with zigzag boundaries (e.g., the mesh in Fig.3(d)), which will affect the results of gap triangulation. Therefore, before triangulating the gap, the boundaries are smoothened (e.g., see Fig.3(e)).

The optimal triangulation problem has been well defined in [50]. Given two piecewise linear curves C_P and C_Q with m and n vertices, a boundary bridge triangulation (BBT) is defined as an ordered collection of triangles $M = \{T_1, T_2, \ldots, T_N\}$. For two piecewise linear curves C_P and C_Q with m and n vertices, respectively, there are a total of

$$\binom{m+n-2}{m-1} = \binom{m+n-2}{n-1} = \frac{(m+n-2)!}{(m-1)!(n-1)!}$$

distinct boundary bridge triangulations. Find one particular BBT from this huge pool that will optimize a certain given objective by exhaustive search is neither plausible nor practical. To solve this problem, authors in [50] converted this combinatorial search problem into a shortest path problem, which can be computed by the Dijkstra's algorithm.

When applying this technique to compute an optimized gap triangulation, two problems needs to be solved: 1) what is the property measured on the graph link (i.e., on the formed triangle), and 2) how to determine the starting vertices. We have the second problem because that the piecewise linear curves here are closed while the ones in [50] are open. The first problem is solved by a heuristic that we wish to have resultant triangles as regular as possible. Thus, the following metric is employed to measure the quality of a triangle T

$$J = \left| 2\sqrt{3} \times r / l - 1.0 \right|, \tag{3}$$

where r is the radius of T's inscribed circle, and l is the maximum length of three edges on T. The more regular the triangle T is, the smaller value is given on the metric J. We solve the second problem by another heuristic to select the closest two vertices on C_P and C_Q as the starting vertices. Figure 3(f) shows an example of triangulated gaps filled with quality optimized triangles.

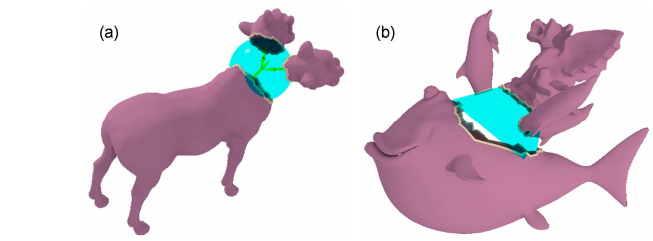

Fig.5. Transient primitives are used to help place component models under composition: (a) the sphere and (b) the truncated cone in blue.

3.4 Transient Surface Remeshing

So far we have obtained the fully tessellated transient mesh surface which seamlessly stitches the compositing components together as shown in Fig.3(f). An *Area Equalizing Remeshing* procedure akin to [4] is conducted to further improve the quality of resultant triangles. Given a target edge length L, the following steps were performed repeatedly for about 5 runs:

- Split edges longer then $4L/3$ at their midpoint;
- Collapse edges shorter than $4L/5$ into their midpoint;
- Flip edges to minimize the deviation from valence 6;
- Iteratively move each vertex p_i to its area-weighted centroid g_i for 10 times by: $p_i \leftarrow p_i + \lambda(I - n_i n_i^T)(g_i - p_i)$ with n_i be the normal vector of p_i and $\lambda = 0.1$ be a damping factor used to avoid oscillations.

Only the edges and vertices not on the boundary of transient surface are processed here. Figure 3(g) shows an example transient mesh surface remeshed from the one given in Fig.3(f).

4 Other Algorithms for Mesh Composition

Other algorithms to support the implementation of mesh compositions are presented in this section. Although they are with minor technical contributions, they are essential components to support our mesh composition framework.

4.1 Primitive Based Components Placement

As mentioned in other mesh composition papers (ref. [13] and [43]), the placement of model components under merging is a very tedious job which needs a lot of interactivities. To reduce this burden, we developed a primitive based components placement scheme. After scissoring the models using the method in [14], the boundary on one model component is firstly selected. A selected primitive is then placed on the selected boundary – the primitive could be a cube, a cylinder, a sphere, or a truncated cone. After that, other components can be placed, rotated and translated on the surface of the primitive. Meanwhile, the dimensions of the primitive can also be adjusted so that the position of placed components are changed automatically by keeping its relative local coordinate with the transient primitives. With these primitives, it is easier to place the models under composition. Figure 5 shows examples using two different primitives – a sphere and a truncated cone. Note that introducing these primitives is only to reduce the work-loading for placing the components under composition. Using other techniques (e.g., [43]) to place the components will not affect the composition technique presented above.

4.2 Geometry Detail Propagation

The transient mesh surface produced by tessellating the trimmed implicit surface lacks of geometry details. Two methods have been implemented in our framework so that can add geometry details in post-processing steps, which can be considered as a 3D extension of the image completion in [46] akin to [54].

Signals for geometry details

The difference between a surface S with geometry detail and its shape after low-pass filtering (i.e., $\bar{S}$) can be considered as signals. Directly encoding/decoding this difference in Euclidean space is named as the distance-map method, which always leads to unexpected distortions. Thus, the authors in [45] adopted the differences of Laplacian coordinates on S and $\bar{S}$ as the signal for geometry details. More specifically, for a vertex $v_i \in S$, the signal is defined as $\xi_i = \delta_i - \bar{\delta}_i$ where δ_i and $\bar{\delta}_i$ are the Laplacian coordinates of v_i on S and $\bar{S}$. To overcome the distortion of Laplacian operator on irregular meshes, we adopt the cotangent weighted Laplacian operator but not the uniform Laplacian operator in [45]. The cotangent weighted Laplacian coordinate is consistent with the curvature flow surface fairing method in [9], which is employed here to generate the smooth surface $\bar{S}$ by the curvature flow operator. After obtaining $\bar{S}$, the geometry detail signal is recorded by encoding $\xi_i = \delta_i - \bar{\delta}_i$.

Reversely, once the geometry detail signal ξ_i is given on every vertex i of a smooth surface $\bar{S}$, we first compute its corresponding Laplacian coordinate $\bar{\delta}_i$ using the current position of vertices. δ_i can then be calculated by $\delta_i = \bar{\delta}_i + \xi_i$. Lastly, we re-compute the position of every vertex by their curvature flow Laplacian coordinates (i.e., δ_i) through a Least-Square fitting system (ref. [45]). Based on this geometry detail encoding/decoding method, two schemes for propagating details on the smooth transient mesh surface are developed.

Structured geometry detail propagation

The structural geometry details are propagated along a user specified curve patch by patch. Firstly, patches are constructed along the propagation governing curve. After drawing a governing curve, it is resampled into m anchor points and projected onto the surface $\bar{S}$ (i.e., the smoothed one). The intersection point between the curve and the boundary of the transient surface needs to be determined (e.g., the green point in Fig.6(c)), which is named as the *bound anchor points*. By the bound anchor points, the resampled points are grouped into two types: 1) the anchor points fall in the region with geometry detail (e.g., the red points in Fig.6(c)), and 2) the anchor points in the region without geometry detail (e.g., the blue ones in Fig.6(c)). We create rectangular domains centered at every anchor point. On the tangent plane of an anchor point, two rectangles are defined. The width of the inner rectangle (the red rectangle in Fig.6(c)) along the governing curve is 2 times the distance between the two neighboring anchor points, and the width perpendicular to the curve is related to the bandwidth of the structured geometry details and is defined by users. The width of the outer rectangle (the green one in Fig.6(c)) is 1.5 times of the inner rectangle width. By [11], the triangles

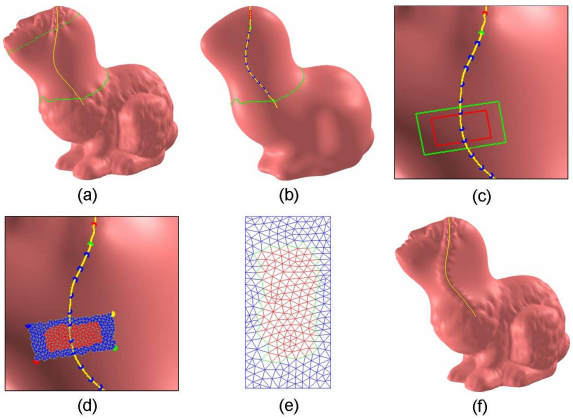

Fig.6. Structured geometry detail propagation : (a) user specified propagation governing curve, (b) the surface is sampled and projected onto the smoothed model – the intersection anchor point between the curve and the boundary of transient surface is computed, (c) two rectangles are constructed on the tangent plane of every anchor point, (d) the triangles fall in different rectangles are grouped, (e) the parameterization result by [11], and (f) the reconstructed shape by the structured geometry detail.

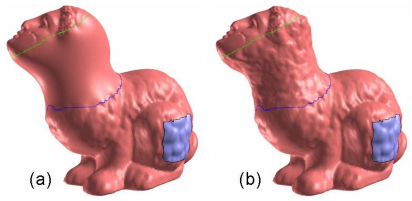

Fig.7. Unstructured geometry detail propagation: (a) the model before detail reconstruction – the source region for geometry detail exemplars are selected in blue, and (b) the reconstructed model with geometry details.

whose projections to the tangent plane fall in the outer square can be parameterized into a rectangular domain (see Fig.6(e)). Only the triangles fall in the inner rectangle show small distortions on the parameterization (i.e., the red triangles in Fig.6(d) and 6(e)). Other triangles (e.g., the blue ones in Fig.6(d)) act as buffers to absorb distortions, which is similar to [55]. Also, only the vertices in the inner rectangle are considered in our structure propagation algorithm.

Starting from the patch associated with the bound anchor point, we fill geometry details to the vertices in the inner rectangle of each patch. More specifically, for a patch Ψ_Q, we search among all patches associated with the anchor point in the region with geometry detail (i.e., the red ones in Fig.6(c)), and select the patch Ψ_P that gives the minimal value on the following difference error

$$E = \sum_{i \in Q} \left\| \xi_i - T\xi_{P(i)} \right\| / |Q| . \quad (4)$$

In Eq.(4), Q denotes the set of vertices in Ψ_Q with geometry detail assigned, $\xi_{P(i)}$ represents the geometry detail at the corresponding point $P(i)$ of vertex $v_i \in \Psi_Q$ in patch Ψ_P, and T is the transformation matrix between two local frames. In general, $P(i)$ shall not be a vertex so that its geometry detail is computed by the barycentric interpolation of details on vertices of the triangle holding $P(i)$. After finding the best match patch Ψ_P of Ψ_Q, all vertices in Ψ_Q without geometry detail are assigned by the detail on its corresponding point on Ψ_P. In this way, we can progressively assign geometry details to the vertices in the inner

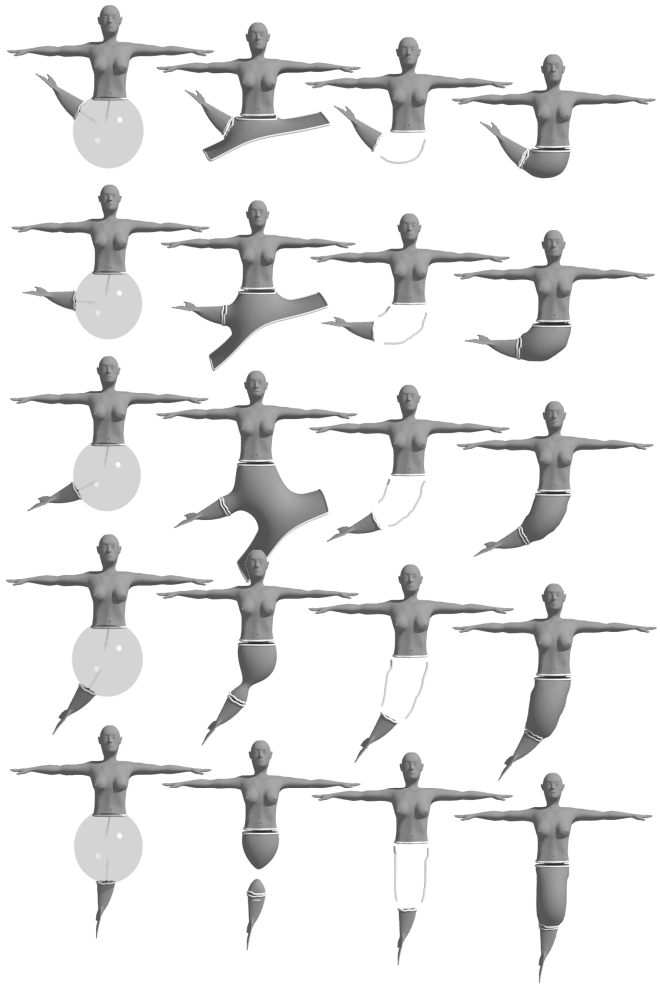

Fig.8. Comparison between the transient implicit surfaces modeled without silhouette constraints (the 2nd column) versus with silhouette constraints (the 3rd and 4th columns) under different orientations: from top to bottom, the angle between two components varies from 60, 90, 120, 150 to 180 degrees respectively.

square of all patches along the propagation curve and then reconstruct their geometry. Figure 6 gives such an example.

Unstructured geometry detail propagation

To define a mesh-free geometry similarity metric, similar to [53], we build a geodesic fan with eight branches of the length $1.5\bar{L}_{\max}$ ($\bar{L}_{\max}$ is the maximal edge length) on a vertex. Let $p_{i,j}$ and $q_{i,j}$ denote the sampling points on the geodesic fans at two vertices p and q respectively, where i (i=0,..,7) is the branch index and j (j=0,…,2) is the index of sample points on a branch. The similarity of geometry details on p and q is defined as

$$\Pi(p,q) = \min_k \{ \frac{1}{\sum_i \sum_j \tau_{i,j}} \sum_i \sum_j \tau_{i,j} L^2(\xi(p_{i,j}), \xi(q_{((i+k)\%8),j})) \} \quad (5)$$

where $\tau_{i,j}$ is the weight of sampling point on the geodesic fan. $\tau_{i,j} = 1$ when $p_{i,j}$ is in a triangle whose vertices are all with ξ assigned; otherwise, $\tau_{i,j} = 0$. $L^2(\xi_p, \xi_q)$ is the L^2-norm of the difference between two geometry detail vectors after rotating them to have the same surface normal direction. For a surface point p inside a triangle, its geometry detail $\xi(p)$ again is calculated by the barycentric interpola-

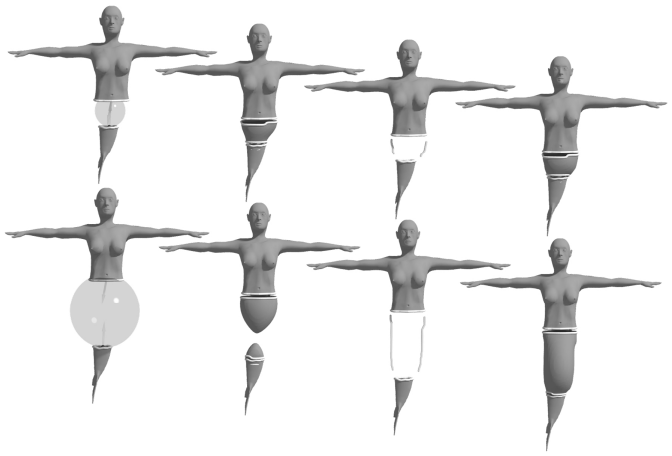

Fig.9. Comparison between the transient implicit surfaces modeled without (the 2nd column) versus with silhouette constraints (the 3rd and 4th columns) under different distances.

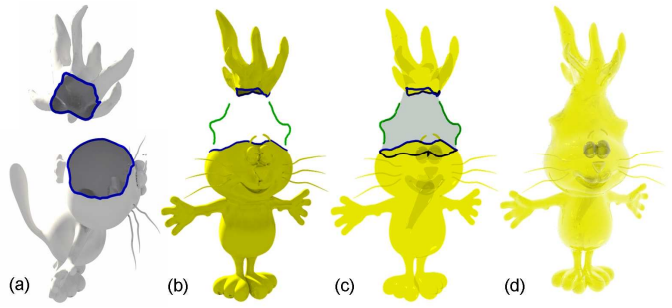

Fig.10. A cartoon cat is created by composing two components (a) where the boundary openings are non-planar. To better control the shape of transient surface, two strokes in green have been added to specify the silhouettes (b). The constructed transient surface (c) and the final result (d) satisfy the user specified silhouettes.

tion. The similarity definition here is different from the one used in the structured propagation, where no rotation is allowed.

The value of $\Sigma_i \Sigma_j \tau_{i,j}$ at a vertex gives the confidence factor of geometry detail – the larger the value, the more confident is the vertex. Our unstructured geometry detail propagation employs this confidence factor to determine the filling order of vertex geometry detail. Specifically, among all vertices without geometry detail, the vertex v_p with the highest confidence is chosen. The vertex v_q showing the highest similarity to v_p in the user specified source region (e.g., the blue ones in Fig.7(a)) is selected, and the geometry detail $\xi(v_q)$ is then assigned to v_p. After that, the confidence factors and the geodesic fans of vertices adjacent to v_p are updated, and the search and filling steps are repeated until the geometry detail of all vertices in the transient region are assigned. The strategy employed here is similar to the image completion algorithm in [8]. Note that different regions could choose different source regions as templates to fill in geometry details. Lastly, the surface mesh with details is reconstructed by the filled geometry details (e.g., Fig.7(b)).

5 Experimental Results and Discussion

We have implemented the proposed approach on a PC with Intel Pentium IV 2.4GHz CPU + 512M RAM. The following two tests are chosen to study the implicit surface based mesh composition.

- The first test is to investigate the shape change of RBF-based implicit surface with the orientation variation of merging boundaries on the models under composition. For the sea-maids constructed in Fig.8, we progressively increase the orientation of the tail staring from 60 degree to 90, 120, 150, until 180 degree. The transient surfaces modelled without silhouette constraints are with very poor shape (as shown in the second column in Fig.8). However, after adding silhouette constraints, we can easily control its shape well (as shown in the fourth column in Fig.8).
- The second test is similar to above but with the distance variation (see Fig.9). With the increase of distance while keeping the same boundary conditions (both position and normal constraints on openings), the transient RBF surface without silhouette constraints becomes more and more narrow in the sense of pulling an elastic object (e.g., see the second column in Fig.9). The result can be improved by adding silhouette constraints (see the fourth column in Fig.9).

Besides the three-head spirit model shown in Fig.1, some more examples can be found in Figs.10-12. Figure 10 shows a mesh composition example with non-planar boundary openings. Silhouettes are specified in Fig.10(b) to design the shape of the transient surface. Different from those approaches that fully convert models into implicit representation like [29] and [30], here only the transient part is an implicit surface – therefore, the details on original models are retained. In Fig.11, two legs of an elephant are replaced by the legs of amadilo. Details are reconstructed on the smooth transient surface. The last example shown in Fig.12 is employed to show another benefit of our implicit transient surface based mesh composition approach – the ability of topology variation. As shown in Fig.12(a), the default merging result from a bottle and a torus is a mesh model with genus-1 topology. However, after adding a stroke to specify the silhouette of a hole in the middle of the transient surface, our mesh composition framework can create a more complex transient surface to merge two components into a model with genus-2 topology. This is a function that has not been provided in other mesh composition approaches. Although some sketching approaches provide such functions (like [31]), their method is based on a 2D triangulation step and a mesh inflating step which cannot be applied to the mesh composition directly.

When using the LU-GPU library to compute the RBF surface fitting and our localized MC algorithm to tessellate the transient surface, the smooth mesh composition step can always be finished in less than 10 seconds (i.e., at interactive speed) on our implementation. For the post-processing of geometry detail propagation (which is an optional step), structure propagation of detail signals can also be finished in an interactive speed; however, the unstructured propagation takes much longer time – usually in few minutes as it involves many times of global search. The detail mesh surface reconstructed from assigned geometry details can be finished in less than 10 seconds by using the TAUCS solver [47].

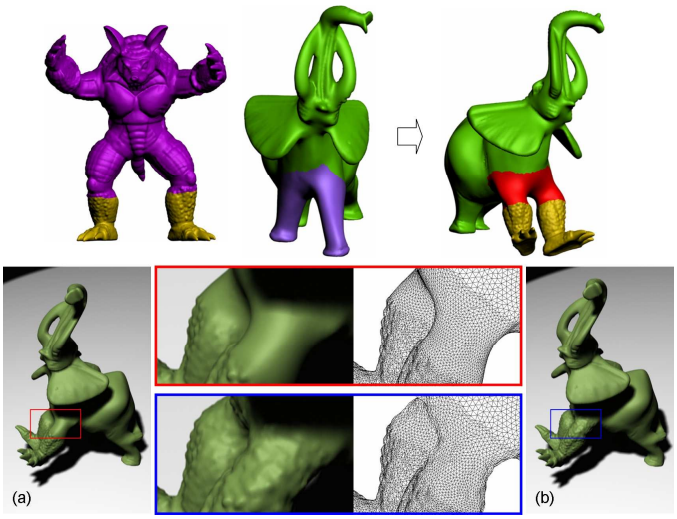

Fig.11. A mesh composition example with unstructured geometry detail propagation – an elephant has its legs replaced by the amadilo's: (a) the model before detail reconstruction and (b) the reconstructed model with geometry details.

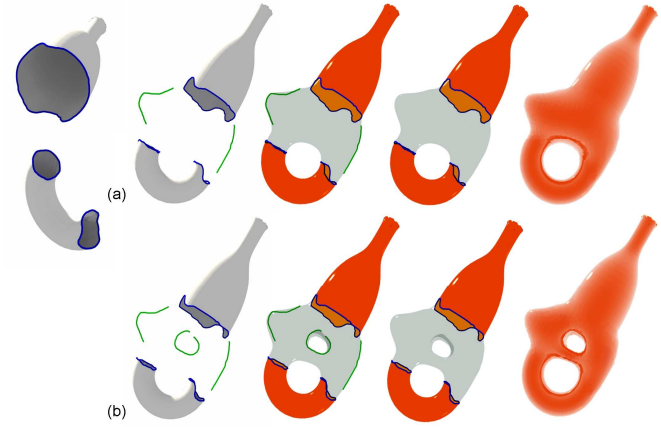

Fig.12. The topology of transient surface can be easily changed by adding more silhouette curves: (a) the default composition result from a bottle and a torus is a mesh model with genus-1 topology, and (b) after specifying more silhouettes, our approach can create a more complex transient surface to merge two components into a model with genus-2 topology.

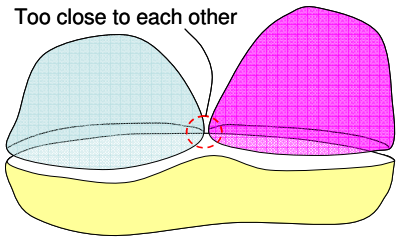

Fig.13. If two merging boundaries are too close to each other, a gap with more complex topology than strip will be formed.

5.1 Limitations

The current implementation of our mesh composition framework shows several limitations. The first limitation comes from the localized Marching Cubes algorithm. The third step of our localized MC algorithm assumes that the left gap to be filled by an optimal triangulation is in the topology of a strip. However, this is not *true* in the following extreme case. If two of the boundary openings are very close to each other so that some parts of them fall in the same cube, we then need to triangulate a gap with one ring on one side while two rings on the other side (as shown in Fig. 13). This is a more complex problem than computing an optimal strip triangulation. To avoid this, in our current implement, we require the placement of models should let the distance between any two merging boundaries be greater than $\sqrt{3}w$. Here, w is the width of cubes adopted in the localized MC algorithm. One of our future works is planned to develop a constrained Delaunay triangulation method in 3D to triangulate the gap region with more complex topology.

Secondly, our implementation assumes that every stroke generates a silhouette curve on a plane linking to the endpoints of two openings. This is relatively simple. Although we can further adjust the shape of silhouettes on the projection plane and the depth plane later (e.g., the silhouettes to specify holes in Fig.12 are generated by this simple extension), a more complex interface is wanted by which one silhouette can be specified by several strokes as what is used in charcoal drawings. We did not include this work in this paper because we consider this as minor technical contribution. This is considered as another future work.

Thirdly, the geometry detail encoding/decoding method presented in this paper works well for the rilievo-like details, which can be complex in shapes but with simple topology; however, the Laplacian coordinate based detail encoding/decoding fails for details with complex topology such as bowknot. The method for constructing details with complex topology on the smooth transient surface will be investigated in our future research, where the methods of volume texture [37] and [39] will be considered.

Geodesic fans with different lengths lead to different similarity comparison results – greater support size is more robust but can hardly distinguish sharp features, while smaller support size works well on sharp features but are violated by local normal disturbances. There is no simple means to choose an appropriate support size for the geodesic fans. We choose the size through experiments. As shown in the tests of Fig.14, different similarities are shown on the bunny model with the geodesic fan's lengths as 0.005, 0.01, 0.015 and 0.02 of the diagonal length of the bunny's bounding box. The similarities obtained by the geodesic fans with 0.01 diagonal length are the best. Therefore, we employ this length for geodesic fans in all our examples.

Lastly, the current implementation does not consider about the intersection between the transient implicit surface and the existing meshes under composition. We rely on users to find the intersection and change the shape of transient surface by silhouette curves. Although such case rarely occurs in our tests, it cannot be fully prevented. This will be investigated in our future work. Furthermore, keeping the original meshes be rigid may cause problems when the models are not well aligned and the merging boundaries are very close to each other. For this scenario, the automatic shape adjustment approach as [19] will be helpful.

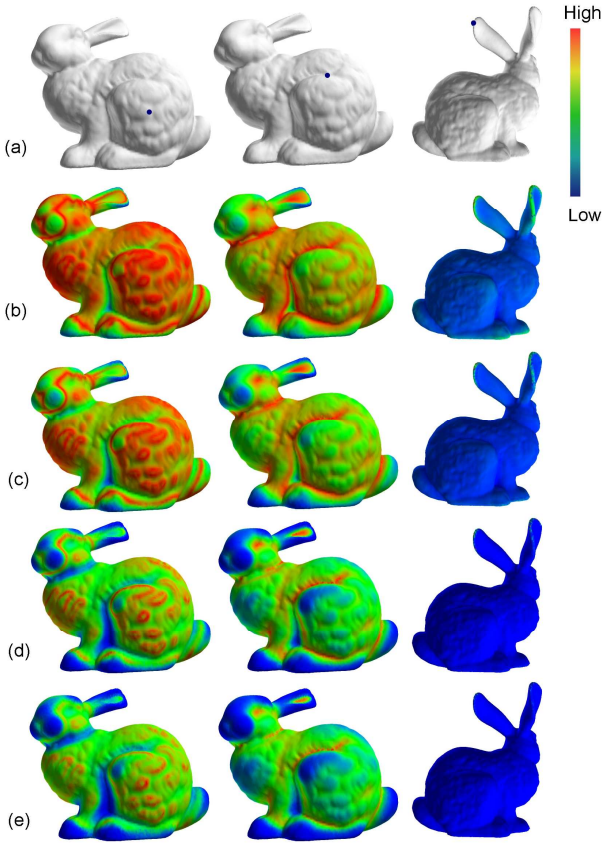

Fig.14. Similarity maps with different geodesic fan length: (a) the similarity of all vertices to the vertices given in blue color is computed, (b)-(e) the geodesic fan's lengths are given as 0.005, 0.01, 0.015 and 0.02 of the diagonal length of the bounding box.

6 Conclusion

In this paper, we developed a novel mesh composition framework for creating 3D models from models with arbitrary boundary topology. The novel mesh composition framework is based on two techniques developed here. After placing the components of merging in their right pose, a RBF-based implicit surface is adopted to smoothly interpolate the boundaries of models under composition. To achieve better shape control for the transient part, a new interface is developed to control the shape of the implicit transient surface by using sketches to specify the expected silhouettes. A localized Marching Cubes algorithm is investigated to tessellate the implicit transient surface so that the mesh surface of composed model is generated. Based on these two technical contributions, our mesh composition framework can fuse models with arbitrary boundary topology – but all existing mesh composition approaches need to have pairwise merging boundaries. Such an exciting new function provides a method to build more complex models by mesh composition easily and efficiently. Also, some assistant techniques have been presented in this paper to help 1) pose the models under composition more easily and 2) propagate structured and unstructured geometry details on the smooth transient surface. The examples presented in this paper show the success of these functions.

Acknowledgment

The authors would like to thank the AIM@SHAPE Shape Repository for sharing some of the models. Juncong Lin, Charlie C.L. Wang and Kin-Chuen Hui were partially supported by Hong Kong RGC/CERG grant CUHK/412405 and DAG grant CUHK/2050341. Xiaogang Jin is supported by the China 863 program (Grant: 2006AA01Z314), the Natural Science Foundation of Zhejiang Province (Grant: R105431), the National Natural Science Foundation of China (Grant: 60573153) and NCET-05-0519.

References

[1] S. Akkouche, E. Galin, "Adaptive implicit surface polygonization using marching triangles," *Computer Graphic Forum*, vol.20, pp.67-80, 2001.

[2] C. Andujar, P. Brunet, A. Chica, I. Navazo, J. Rossignac, and A. Vinacua, "Optimizing the topological and combinatorial complexity of isosurfaces," *Computer-Aided Design*, vol.37, pp.847-857, 2005.

[3] H. Biermann, I. Martin, F. Bernardini, and D. Zorin, "Cut-and-paste editing of multiresolution subdivision surfaces," *ACM Tran. Graphics*, vol.21, no.3, pp. 312-321, 2002.

[4] M. Botsch, and L. Kobbelt, "A remeshing approach to multiresoluton modeling," *Proc. of the EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 185-192, 2004.

[5] E.V. Chernyaev, "Marching cubes 33: construction of topologically correct isosurfaces," *Technical Report CERN CN-95-17*, 1995.

[6] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum, and T.R. Evans, "Reconstruction and representation of 3D objects with radial basis functions," *Proc. of SIGGRAPH 2001*, pp.67-76, 2001.

[7] U. Clarenz, U. Diewald, G. Dziuk. M. Rumpf, R. Rusu, "A finite element method for surface restoration with smooth boundary conditions," *Computer Aided Geometry Design*, vol.5, pp.427-445, 2004.

[8] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. on Image Processing*, vol.13, no.9, pp.1200-1212, 2004.

[9] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," *Proc. of ACM SIGGRAPH 99*, pp. 317-324, 1999.

[10] M.J. Duerst, "Letters: Additional reference to marching cubes," *Computer Graphics*, vol.22, pp.72-73, 1988.

[11] M. Floater, "Parameterization and smooth approximation of surface triangulations," *Computer Aided Geometric Design*, vol.14, no.3, pp.231-250, 1997.

[12] H. Fu, C.-L. Tai, and H. Zhang, "Topology-free cut-and-paste editing over meshes," *Proc. of Geometric Modeling and Processing*, pp.173-184, 2004.

[13] H. Fu, O.K.C. Au, C.-L. Tai, "Effective derivation of similarity transformations for implicit Laplacian mesh editing," *Computer Graphics Forum*, vol.26, no.1, pp.34-45, 2007.

[14] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, "Modeling by example," *ACM Trans. on Graphics*, vol.23, no.3, pp.652-663, 2004.

[15] N. Galoppo, N.K. Govindaraju, M. Henson, and D. Manocha, "LU-GPU: efficient algorithms for solving dense linear systems on graphics hardware," *Proc. of the 2005 ACM/IEEE conference on Supercomputing*, pp.3, 2005.

[16] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka, "Curvature maps for local shape comparison," *Proc. of Shape Modeling International*, pp.244-256, 2005.

[17] T. Hassner, L. Zelnik-Manor, G. Leifman, and R.Basri, "Minimal-cut model composition," *Proc. of International Conference on Shape Modeling and Applications*, pp.72-81, 2005.

[18] A. Hilton, A.J. Stoddart, J. Illingworth, and T. Windeatt, "Marching triangles: range image fusion for complex object modeling," *Proc. of 3rd IEEE International Conference on Image Processing*, vol.2, pp.381-384, 1996.

[19] X. Huang, H. Fu, O.K.-C. Au, and C.-L. Tai, "Optimal boundaries for Possion mesh merging," *Proc. of ACM Symposium on Solid and Physical Modeling*, pp.35-40, 2007.

[20] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: a sketching interface for 3d freeform design," *Proc. of SIGGRAPH 99*, pp 409-416, 1999.

[21] S. Jin, R.R. Lewis, and D. West, "A comparison of algorithms for vertex normal computation," *The Visual Computer*, vol.21, no.1-2, pp.71-82, 2005.

[22] X. Jin, J. Lin, C.C.L. Wang, J. Feng, and H. Sun, "Mesh fusion using functional blending on topologically incompatible sections," *The Visual Computer*, vol.22. no.4, pp. 266-275, 2006.

[23] T. Ju, "Robust repair of polygonal models," *ACM Trans. on Graphics*, vol.23, no.3, pp.888-895, 2004.

[24] T. Kanai, H. Suzuki, J. Mitani, and F. Kimura, "Interactive mesh fusion based on local 3D metamorphosis," *Proc. of Graphics Interface 99*, pp. 148-156, 1999.

[25] O. Karpenko, J. Hughes, and R. Raskar, "Free form sketching with variational implicit surfaces," *Computer Graphics Forum*, vol.7, no.3, pp. 585-594, 2002.

[26] V. Kraevoy, and A. Sheffer, "Template-based mesh completion," *Proc. of Eurographics Symposium on Geometry Processing*, pp.13-22, 2005.

[27] T. Lewiner, H. Lopes, A.W. Vieira, and G. Tavares, "Efficient implementation of marching cubes' cases with topological guarantees," *Journal of Graphics Tools*, vol.8, pp.1-15, 2003.

[28] W. Lorensen, and H. Cline, "Marching cubes: a high resolution 3D surface construction algorithm," *Computer Graphics*, vol.21, pp.163-169, 1987.

[29] K. Museth, D.E. Breen, R.T. Whitaker, and A.H. Barr, "Level set surface editing operators," *ACM Trans. on Graphics*, vol.21, no.3, pp.330–338, 2002.

[30] K. Museth, D. Breen, R. Whitaker, S. Mauch and D. Johnson, "Algorithms for interactive editing of level set models," *Computer Graphics Forum*, Vol.24, No.4, pp.821-841, 2005.

[31] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "FiberMesh: designing freeform surfaces with 3D curves," *ACM Trans. Graph*, vol.26, no.3, 2007.

[32] M.X. Nguyen, X. Yuan, and B. Chen, "Geometry completion and detail generation by texture synthesis," *The Visual Computer*, vol.21, no.8-10, pp.669-678, 2005.

[33] P. Ning, and J. Bloomenthal, "An evaluation of implicit surface tilers," *IEEE Computer Graphics and Applications*, vol.13, pp.33-41, 1993.

[34] F.S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Trans. on Visualization and Computer Graphics*, vol.9, no.2, April-June 2003, pp.191-205.

[35] S. Park, X. Guo, H. Shin, and H. Qin, "Shape and appearance repair for incomplete point surfaces," *Proc. of Tenth IEEE International Conference on Computer Vision (ICCV'05)*, vol.2, pp.1260-1267, 2005.

[36] G. Pasko, and A. Pasko, "Trimming implicit surfaces," *The Visual Computer*, vol. 20, no.7, pp.437-447, 2004.

[37] J. Peng, D. Kristjansson, and D. Zorin, "Interactive modeling of topologically complex geometric detail," *ACM Trans. on Graphics*, vol.23, no.3, pp.635-643, 2004.

[38] J. Podolak, S. Rusinkiewicz, "Atomic volumes for mesh completion," *Proc. of Eurographics Symposium on Geometry Processing*, pp.23-32. 2005.

[39] S.D. Porumbescu, B. Budge, L. Feng, and K.I. Joy, "Shell maps," *ACM Trans. on Graphics*, vol.24, no.3, pp.626-633, 2005.

[40] B. Schmitt, A. Pasko, G. Pasko, and T. Kunii, "Rendering trimmed implicit surfaces and curves," *Proc. of AFRIGRAPH 2004*, pp.7-13, 2004.

[41] J. Schreiner, C. Scheidegger, and C. Silva, "High-quality extraction of isosurfaces from regular and irregular grids," *IEEE Trans. on Visualization and Computer Graphics*, vol.12, no.5, pp.1205-1212, 2006.

[42] A. Sharf, M. Alexa, and D. Cohen-Or, "Context-based surface completion," *ACM Trans. on Graphics*, vol.23, no.3, pp.878-887, 2004.

[43] A. Sharf, M. Blumenkrants, A.Shamir, and D. Cohen-Or, "SnapPaste: an interactive technique for easy mesh composition," *The Visual Computer*, vol.22, no.9, pp.835-844, 2006.

[44] K. Singh, and R. Parent, "Joining polyhedral objects using implicitly defined surfaces," *The Visual Computer*, vol.17, no.7, pp.415-428, 2001.

[45] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Rossl, and H.P. Seidel, "Laplacian surface editing," *Proc. of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp.179-188, 2004.

[46] J. Sun, L. Yuan, J. Jia, and H.Y. Shum, "Image Completion with structure propagation," *Proc. of SIGGRAPH 2005*, pp. 861-868, 2005.

[47] S. Toledo, *TAUCS: A Library of Sparse Linear Solver*, version 2.2, 2003. Tel-Aviv University, http://www.tau.ac.il/stoledo/taucs/.

[48] G. Turk, and J.F. O'brien, "Modeling with implicit surfaces that interpolate," *ACM Trans. on Graphics*, vol.21, no.4, pp.855-873, 2002.

[49] J. Verdera, V. Caselles, M. Bertalmio, G. Sapiro, "Inpainting surface holes," *Proc. of International Conference on Image Processing*, pp.14-17. 2003.

[50] C.C.L. Wang, and K. Tang, "Optimal boundary triangulations of an interpolating ruled surface," *Journal of Computing and Information Science in Engineering*, *ASME Transactions*, vol.5, no.4, pp.291-301, 2005.

[51] S. Yoshizawa, A. Belyaev, and H.P. Seidel, "Smoothing by example: mesh denoising by averaging with similarity-based weights," *Proc. of Shape Modeling International*, pp.38-44, 2006.

[52] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, and H.Y. Shum, "Mesh editing with poisson-based gradient field manipulation," *ACM Trans. Graphics*, vol.23, no.3, pp.644-651, 2004.

[53] S. Zelinka, and M. Garland, "Similarity-based surface modeling using geodesic fans," *Proc. of 2nd Eurographics Symposium on Geometry Processing*, pp.209-218, 2004.

[54] S. Zelinka, and M. Garland, "Surfacing by numbers," *Proc. of Graphics Inerface*, pp107-113, 2006.

[55] E. Zhang, K. Mischaikow, and G. Turk, "Feature-based surface parameterization and texture mapping," *ACM Trans. Graphics*, vol. 24, no. 1, pp.1-27, 2005.

[56] Q.-Y. Zhou, T. Ju, and S.-M. Hu, "Topology repair of solid models using skeletons," *IEEE Trans. on Visualization and Computer Graphics*, accepted, 2007.

Juncong Lin is a PhD candidate of the State Key Lab of CAD&CG, Zhejiang University. He received his BSc degree in Environmental Engineering in 2003 from Zhejiang University. His research interests include mesh editing and modelling.

Xiaogang Jin is a professor of the State Key Lab of CAD&CG, Zhejiang University. He received his BSc degree in computer science in 1989, MSc and PhD degrees in applied mathematics in 1992 and 1995, all from Zhejiang University. His current research interests include implicit surface computing, special effects simulation, mesh fusion, texture synthesis, crowd animation, cloth animation and facial animation.

Charlie C. L. Wang is currently an Assistant Professor at the Department of Mechanical and Automation Engineering, the Chinese University of Hong Kong. He gained his B.Eng. (1998) in Mechatronics Engineering from Huazhong University of Science and Technology, M.Phil. (2000) and Ph.D. (2002) in Mechanical Engineering from the Hong Kong University of Science and Technology. He is a member of IEEE and ASME. His current research interests include geometric modeling in computer-aided design and manufacturing, biomedical engineering, and computer graphics, as well as computational physics in virtual reality.

Kin-Chuen Hui received his B.Sc and Ph.D in Mechanical Engineering in 1979 and 1990 respectively from the University of Hong Kong. Before joining the Chinese University of Hong Kong in 1992, he was a consultant in the CAD Services Centre of the Hong Kong Productivity Council. He is currently a Professor of the Mechanical and Automation Engineering Department at the Chinese University of Hong Kong, and is the director of the Computer-Aided Design Laboratory. He is a member of the editorial board of the Journal of Computer-Aided Design. His research interests include computer graphics, geometric and solid modeling, virtual reality, and their applications.