Hardware-accelerated visualization of volume-sampled distance fields

Shuntaro Yamazaki
University of Tokyo, RIKEN
shun@cvl.iis.u-tokyo.ac.jp

Kiwamu Kase
RIKEN
kiwamu@riken.go.jp

Katsushi Ikeuchi
University of Tokyo
ki@cvl.iis.u-tokyo.ac.jp

Abstract

We present a method of visualizing volume-sampled distance fields, taking advantage of hardware-acceleration of modern graphics hardware. Although conventional distance fields can represent only 2-manifold surfaces in a stable way, we have developed a technique of representing both manifold and non-manifold surfaces in a volume created by sampling a segmented distance field. The volume-sampled distance fields can be visualized effectively with pre-integrated volume rendering by embedding the interpolating function as a lookup table called vertex generation diagram into graphics hardware. We developed a simple system for smoothly blending two surface models in the distance function domain, and confirmed that our proposed representation of surface models is applicable to existing methods of geometric processing using implicit representations. We also confirmed that proposed methods of deformation and visualization can be executed at sufficient quality and speed using commodity graphics hardware on a standard PC.

1 Introduction

Commonly-used representations for surface models in computer graphics are composed of *parametric surfaces* and *implicit surfaces*. While an implicit representation of surfaces can be converted into a parametric representation by existing methods of polygonization, it is difficult to convert them in reverse; that is, it is not necessarily clear how to generate the function whose isocontour corresponds given to parametric surfaces. With a function which returns the distance to given surface, the surface can be defined as 0-level sets of the function. Although the computation of distance, called the *distance transformation*, is usually feasible, the function which spans the distance field is often too complicated to be obtained in functional form.

One approach to this problem is representing the function as a set of sampled points, instead of the function itself. The surface is defined as a set of sampled values and stored in a volumetric data structure, such as a volume or an octree. Although the values of a distance field are discretely defined by sample points, we can obtain an approximate distance field by interpolating these values. In this sense, the volume-sampled distance field can be regarded as a kind of implicit representation of the surface.

The limitation of conventional implicit surfaces is that they can deal only with 2-manifolds in a stable way because the corresponding implicit fields are usually defined by real-valued functions which bisect space into interior and exterior. A typical example based on the function is the *signed distance field*, where the value assigned to each point in the field is a real-valued distance with positive sign when the point is outside the region enclosed by the surface, and with negative sign when the point is inside the surface. The surface models represented by a signed distance field must have incorrect topology and geometry

if the surface has boundaries or intersections, where the space cannot be bisected. These cases can occur in non-manifold surfaces.

According to a result of the implicit function theorem, isocontour of a scalar function on non-regular values can be non-manifold. However, the non-manifold isocontour on such values is not guaranteed to be planar, but can be 0-, 1- or 3-manifold. Moreover, the non-manifold features based on non-regular values appear or disappear even by slightly changing of the value, which is not desirable in processing the shape of models.

Recently, we proposed a method of implicitizing and polygonizing non-manifold surfaces using an effective scheme in which the field is represented by the *segmented distance* whose domain is segmented into several regions according to the topology of a given surface, and defined only within certain intervals [14]. The concrete surfaces can be extracted from the segmented distance field through polygonization. In this paper, we extend our previous work to visualization and propose a technique based on volume rendering using hardware-accelerated pixel shading.

Our proposed method can be regarded as a specialization of the method proposed by Engel et al. [5], for implicit surface rendering. We embedded a lookup table, called vertex generation diagram, for interpolating distance values in the graphics hardware as a 2D texture, and visualize the surface using the programmable pixel shader available on the nVidia's GeForce3 graphics chip. Contrary to other existing methods of rendering implicit surfaces, our proposed method can render both manifold and non-manifold surfaces at a sufficient frame rate, independent of the complexity of the geometry or topology of the surface.

This paper first discusses previous related work, then explains the basic concept of implicit surfaces and extends it to non-manifold surfaces. It then describes the methods of volume rendering and shows that they are suitable for rendering implicit surfaces. After presenting some results of rendering, conclusions on the proposed method are given.

2 Related work

2.1 Representation of implicit surfaces

The implicit surface is an area of geometric modeling that has been studied widely by many researchers [1]. Applications range over such fields as modeling, deformation, animation, blending and rendering.

Volume-sampled distance fields can be regarded as a variation of implicit functions. Frisken et al. [6] proposed adaptive sampling of signed distance fields and showed that this has sufficient power to represent details on a surface. Kobbelt et al. [8] proposed a method of representing sharp features on a surface in a volume-sampled distance field by applying a directed distance function. Their work also mentions how to convert a surface from parametric form to implicit form, but their methods are applicable only to manifold surfaces.

Non-manifold surfaces can be modeled in the framework of implicit surfaces by introducing multiple classifications of implicit fields as pointed out in [**?**, 9, 10]. Based on the discussion, Bloomenthal et al. [2] proposed a concrete method of polygonizing non-manifold implicit surfaces using multiple regionalization. Hege et al. [**?**] and Bonnell et al. [3] proposed methods of extracting non-manifold interfaces between some materials in a volume. We also proposed a method of implicitizing arbitrary parametric surfaces by the distance transformation taking account of the discontinuity of the field function [14]. Although these methods are based on theoretically equivalent structures, ours can be effectively implemented, by tabularizing the most time-consuming process of interpolating sampled values into a lookup table, and is therefore suitable for real-time visualization.

2.2 Visualization of implicit surfaces

While implicit surfaces can be visualized in high quality using ray-tracing, this method is not suitable for the interactive visualization since it takes a long time to trace all rays shot from the screen through three-dimensional space. Perry et al. [11] proposed adaptive sampling of distance fields to accelerate rendering; however, the speed of rendering depends on the complexity of the surface and it is not fast enough to render surfaces in interactive frame rates in most cases.

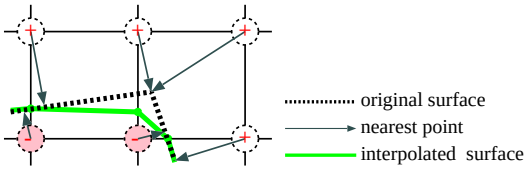

Figure 1: Isosurface reconstructed from sampled values in signed distance field.

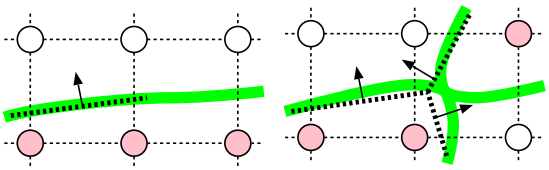

Figure 2: Non-manifold surfaces represented by implicit surfaces using the existing definition.

Witkin et al. [13] proposed a unique method of visualizing surfaces by distributing small particles on the isosurface of the implicit function. The viewer, however, must infer the shape of the implicit surface from the position and orientation of particles. Moreover, the method requires continuity or uniqueness of surface connection which is not present in non-manifold implicit surfaces.

Once implicit surfaces are converted into parametric surfaces such as triangular meshes, they can be easily and effectively visualized by taking advantage of the acceleration of modern graphics hardware. Although many works [2, 14, 7, **?**] have been done on generating a non-manifold polygonal mesh from a volume, they are not suitable for real-time visualization since extracting parametric surfaces before rendering is required as a preprocessing step. This is particularly a problem when the shape of implicit surfaces varies in real-time, or has to be changed interactively.

3 Non-manifold implicit surface

3.1 Implicit surface represented by a volume

Given a surface $S \subset \mathbf{R}^3$, consider the combination of the scalar value t and function f such that

$$p \in S \quad \Leftrightarrow \quad f(p) = t, \tag{1}$$

where $p \in \mathbf{R}^3$ is a point in three-dimensional space. f can be defined as the distance between p and the surface if $t = 0$. The definition of S on the right side of equation (1) is an implicit representation of S, and the surface S defined in this representation is called an *implicit surface*.

Although the definition of f is not unique, a common choice is using the *signed distance*. This function $f(p)$ for a point p returns the Euclidean distance with positive sign when p is outside the region enclosed by S and with negative sign when p is inside S.

Given a parametric surface, the functional form of the distance field is often so complicated that the values of the fields are sampled on a regular grid in three-dimensional space and stored as a volume. The values between grid points are computed by linearly interpolating the values on adjacent points.

A two-dimensional example of the signed distance field represented by a volume is shown in Figure 1. In this figure, an isosurface is reconstructed in two steps; first the points on the isosurface are estimated by linearly interpolating the values on adjacent sample points, then they are linearly connected to obtain concrete isosurfaces.

3.2 Non-manifold surface

In three-dimensional space, a surface S is a 2-manifold if and only if the infinitesimal neighborhood around any point on S is topologically equivalent to a two-dimensional disk. In other words, exactly two different directions can be chosen as the axes of a two-dimensional local coordinate system for any point on the surface. We use the term “manifold” as meaning a 2-manifold in this paper.

Conversely, a surface S is non-manifold if and only if the number of directions which can be chosen on a point on S is less than two, or more than two. Assuming that the infinitesimal neighborhood around every point on S is equivalent to neither a 0-, 1- nor 3-manifold, S is regarded as a non-manifold surface in one of the two following cases.

1. **Surface has boundaries:**

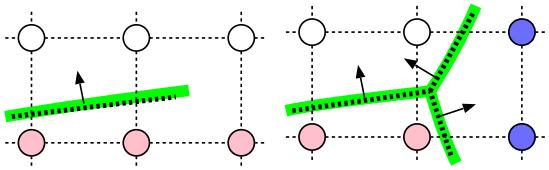

Figure 3: Non-manifold implicit surfaces represented by our proposed method.

There exists a point $p \in S$ around which only one or no direction can be chosen in the local coordinate system. This is formed if the surface has a boundary. An example in two dimensions is illustrated on the left in Figure 2.

2. **Surface has junctions:**
   There exists a point $p \in S$ around which more than two directions can be chosen in the local coordinate system. This is formed if the surface has a junction. An example in two dimensions is illustrated on the right in Figure 2.

As shown in Figure 2, the isosurfaces reconstructed using linear interpolation of the values on adjacent sample points result in topologically wrong shapes.

3.3 Segmented distance field

Two non-manifold features listed in Section 3.2 can be represented as implicit surfaces only by applying multiple classification of the space, instead of two classes of + and −. It is, however, not clear how to create a function whose isocontour represents a given surface. Therefore we start from the conventional signed distance and linear interpolation, then enhance the interpolating function and the distance function in two steps.

3.3.1 Representing boundaries

The boundary of given surface can be represented in implicit form by preventing the isocontour between two sample points from being generated if the signed distances on their points differ by more than the distance between the points in the space. Let u and v be signed distances on adjacent sample points and $u \leq v$, then an isocontour is generated between the points if and only if

$$\begin{aligned} &u \in (-\infty, 0), \\ &v \in [0, \infty), \\ &0 < (-u) + v < w, \end{aligned} \tag{2}$$

where w is the interval of the sample points. When equation (2) holds, the position t of the isocontour between the points is determined by

$$t = \frac{(-u)}{(-u) + v}, \tag{3}$$

where $t \in [0, 1]$ is normalized so as that $t = 0$ and $t = 1$ when the surface is on the point whose value is u and v respectively. Although these equations are all the same as that used in conventional implicit surfaces except for equation (2), they can represent boundaries of the surface as shown on the left in Figure 3.

3.3.2 Representing junctions

The junction of a given surface can be represented in implicit form by applying multiple classifications of the space, instead of two classes of + and −. To accomplish the classification, we define a novel function called the segmented distance, which is essentially composed of two numbers. One is the Euclidean distance to the surface. The other is the number of the region in the space, called the region index. Given a surface S, the region index $r(p)$ for a point p in three-dimensional space is calculated as follows.

1. Divide S into patches S_i along the junction lines if S has any junction.
2. For each S_i, assign divergent index $R^+(i)$ and $R^-(i)$ to the front and back face respectively.
3. For each point p in space, calculate an Euclidean distance $d_E(p)$ to S and a region index $r(p)$ which corresponds to the index of the nearest S_i. If the index is not unique, choose the smallest one. This results in a segmentation of the space.
4. If two infinitesimally distant points have different region indices $r(i)$ and $r(j)$, replace them with another

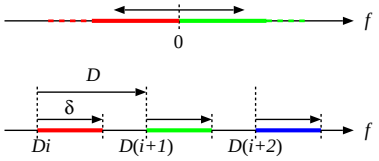

Figure 4: (Top) Signed distance function. (Bottom) Segmented distance function.

unused number so long as $\{r(i), r(j)\}$ has never been $\{R^+(k), R^-(k)\}$ for some k ever before.

5. Iterate step 4 until no more region indices are replaced.

6. Renumber all regions indices from 0 in ascending order.

Generally it is difficult to find the interfaces of the regions in step 4. In our case, since the distance function is sampled only on grid points, the interface can be determined simply by comparing the region index of a grid point with indices assigned to adjacent grid points.

Using $d_E(p)$ and region index $r(p)$, the segmented distance $f(p)$ is defined by

$$f(p) = min(d_E(p), \delta) + Dr(p), \tag{4}$$

where δ and D are positive numbers sufficiently larger than the interval of grid points, and $\delta < D$. Figure 4 shows the comparison between a signed distance function and a segmented distance function.

Given a distance field defined by the segmented distance field, let u and v be signed distances on adjacent sample points and $u \leq v$, then an interface is generated between the points if and only if there exist i and j which satisfy

$$\begin{aligned} u &\in [Di, D(i+1)), \\ v &\in [Dj, D(j+1)), \end{aligned} \tag{5}$$

where $0 \leq i \leq j \leq n-1$ and w is the interval of sample points. The position of the implicit surface is defined as follows.

$$t = \frac{u - Di}{(u - Di) + (v - Dj)}. \tag{6}$$

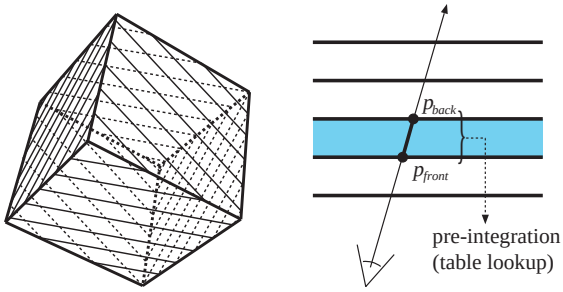

Figure 5: (Left) Slices of a volume. (Right) Pre-integrated volume rendering.

3.3.3 Representing non-manifold features

By combining the definition mentioned above, arbitrary surface models can be represented in a volume-sampled distance field. First the distance field is defined by equation (4) using proper D and δ, then sampled into a volume. Let u and v be segmented distances on adjacent sample points and $u \leq v$, then a surface is generated at the place defined by equation (6), if there exist i, j $(0 \leq i \leq j \leq n-1)$ which satisfy equation (5) and

$$0 < (u - Di) + (v - Dj) < w. \tag{7}$$

4 Interactive visualization

4.1 Pre-integrated volume rendering [5]

The volume-sampled distance fields can be considered as volume data, therefore it can be visualized by ray casting [12], which can be significantly accelerated by sampling the volume as a stack of slices perpendicular to the viewing direction, as shown on the left in Figure 5. The slices of the volume can be efficiently rendered by polygons and textures by taking advantage of graphics hardware acceleration [4].

One major drawback of slice-based volume rendering is that the volume is approximated by a stack of slices, and hence can be sampled densely only on the slices. This means that the volume can be sampled on the slice as densely as possible using interpolation, whereas the sampling in the direction perpendicular to slices is limited to the interval of the slices. This mismatch of sampling density causes aliases in the rendered image.

Recently Engel et al. [5] proposed a method which successfully removes the artifacts by precalculating the color and opacity between slices. In this method, the volume is partitioned into a set of slices as in conventional slice-based volume rendering, and rendered in a slice-by-slice manner from back to front. But when a slice is rendered, one of the adjacent slices is also used to compute the color and opacity taking into account the contribution of the volume between the slices, as shown on the right in Figure 5. Using the volume values on adjacent slices, the volume values between the slices can be estimated by assuming a linear distribution of the volume value. Then the image is rendered accurately taking into account the effects of the volume between the slices.

The integral of color and opacity between slices is calculated numerically. The calculation is called *pre-integration* and can be performed before rendering a volume. By tabularizing the result of pre-integration for all possible pairs of volume values, the computational cost in the volume rendering is reduced drastically. Moreover, using a programmable pixel shader available in common graphics hardware and software, looking-up the table can be implemented effectively taking advantage of fast graphics hardware. Further details on pre-integrated volume rendering are described in [5].

4.2 Tabularizing the interpolating function

Although the pre-integration method is a framework of volume rendering, it is suitable for visualization of implicit surfaces because the surfaces lying between the sampled points are successfully rendered by pre-integration. The segmented distance field for non-manifold surfaces can be rendered effectively by this method since the pre-integration of distance values can be easily performed using equation (6). In out method, the table is used to determine whether and where an iso-surface is generated between two sample points in the segmented distance field; therefore, it is called the *vertex generation diagram*.

An example of vertex generation diagram is shown in Figure 6. Provided that a model is represented as a volume defined by the segmented distance which classifies the field into three regions, as shown on the top, the vertex generation diagram is calculated for six pairs of the regions using equation (6), as shown on the bottom. Because of the limitation of graphics hardware, first we must scale the distance values from 0 to 255. Using $D = 64$ and $\delta = 63$ as the parameters in equation (4), the distance can be defined as $f(p) \in [0, 63]$ where p is in region 1, $f(p) \in [64, 127]$ where p is in region 2, $f(p) \in [128, 191]$ where p is in region 3. When a point marked by a blue cross on the upper hemisphere is rendered, the vertex generation diagram is referenced using two values, one belonging to region 1 (green point), another belonging to region 3 (red point). The horizontal and vertical axes correspond to voxel values on the front slice and the back slice respectively. The diagram maintains both t in equation (6) and color of the point.

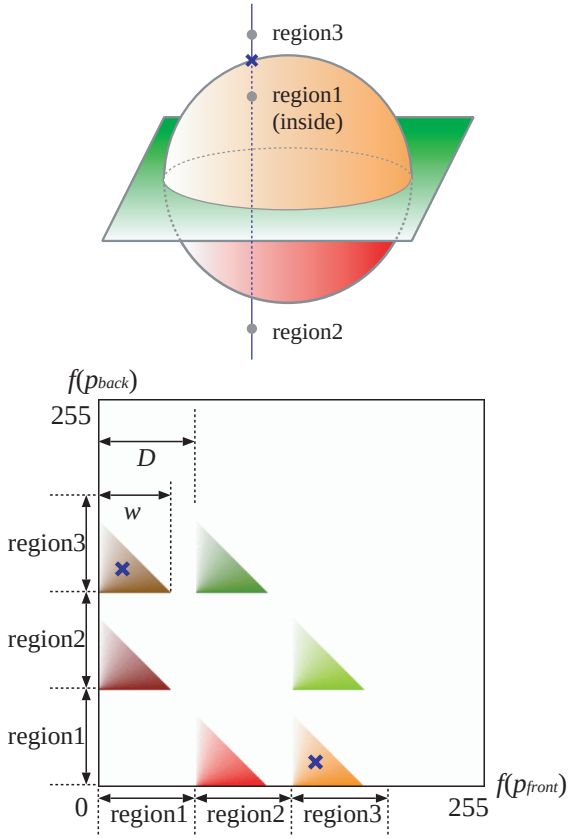

Figure 6: (Top) Example classification of the space. (Bottom) Vertex generation diagram.

When a non-manifold implicit surface is rendered using texture-mapped polygons, the texture image must be

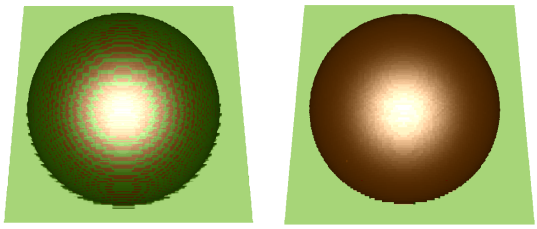

Figure 7: Comparison of texture filtering. (Left) Linear interpolation. (Right) Nearest neighbor interpolation.

Table 1: Performance of rendering

Size of volume	128^3			
Number of slices	32	64	128	256
FPS	>30	>30	25.0	12.3
Size of volume	256^3			
Number of slices	32	64	128	256
FPS	9.9	5.5	2.8	1.5

resampled using nearest-neighbor filter rather than linear filter since the interpolation of segmented distances which belong to different regions can result in a distance that belongs to neither of them. In case a slice lies on the middle of voxels belonging to region 1 and region 3, the obtained field value can belong to neither region 1 nor region 3, but belongs to region 2, if linear interpolation of the distance field is enabled. The undesirable interpolation of field values is illustrated by the blue point in the vertex generation diagram. This results in incorrect surface extraction as shown on the left in Figure 7. This can be prevented by using nearest neighbor interpolation as the texture filter, instead of the linear interpolation. Although disabling linear interpolation of the field can lead to aliasing, pre-integration can interpolate the values between slices, and can improve the quality, as shown on the right in Figure 7.

5 Experimental result

5.1 Rendering non-manifold implicit surfaces

We have implemented the volume rendering based on the method proposed in [5]. All experiments are done on a standard PC equipped with Pentium4 1.7GHz, 1GB of memory and a graphics card containing nVidia's GeForce4 Ti4600.

Figure 8 shows the results of rendering non-manifold implicit surfaces. Given the triangular mesh models, first they are converted into the segmented distance field representation. Then the surface of the field is visualized directly using the hardware-accelerated method proposed in this paper.

Table 1 shows the performance of rendering. A mesh model is converted into 128^3 and 256^3 volumes, then visualized by changing the number of slices.

5.2 Deformation of non-manifold surfaces

To clarify the advantage of the implicit representation of surface models, we did some experiments on a simple metamorphosis of shapes. When surfaces are represented in implicit forms, they can be deformed simply by blending the functions, which can also be performed using graphics hardware. Certainly better correspondences result in better quality of deformation. We however used only simple interpolation of the distance values without any correspondence in these experiments. In addition, we assume the distance fields to be blended are defined by the same segmented distance. Although we believe this scheme can be extended to the metamorphosis of the models defined by different distances, we leave it as future work.

Figure 9 shows the metamorphosis of two different surface models. One has some holes on its surface, but they can be blended smoothly in changing the topology of the surface. The holes on the surface gradually get smaller and finally vanish. Figure 10 shows the metamorphosis between a single surface model and two separate spheres. It is difficult to handle such drastic changes of the topology as shown in these figures when the surfaces are represented in parametric forms. In implicit form, this metamorphosis can be performed only by interpolating two

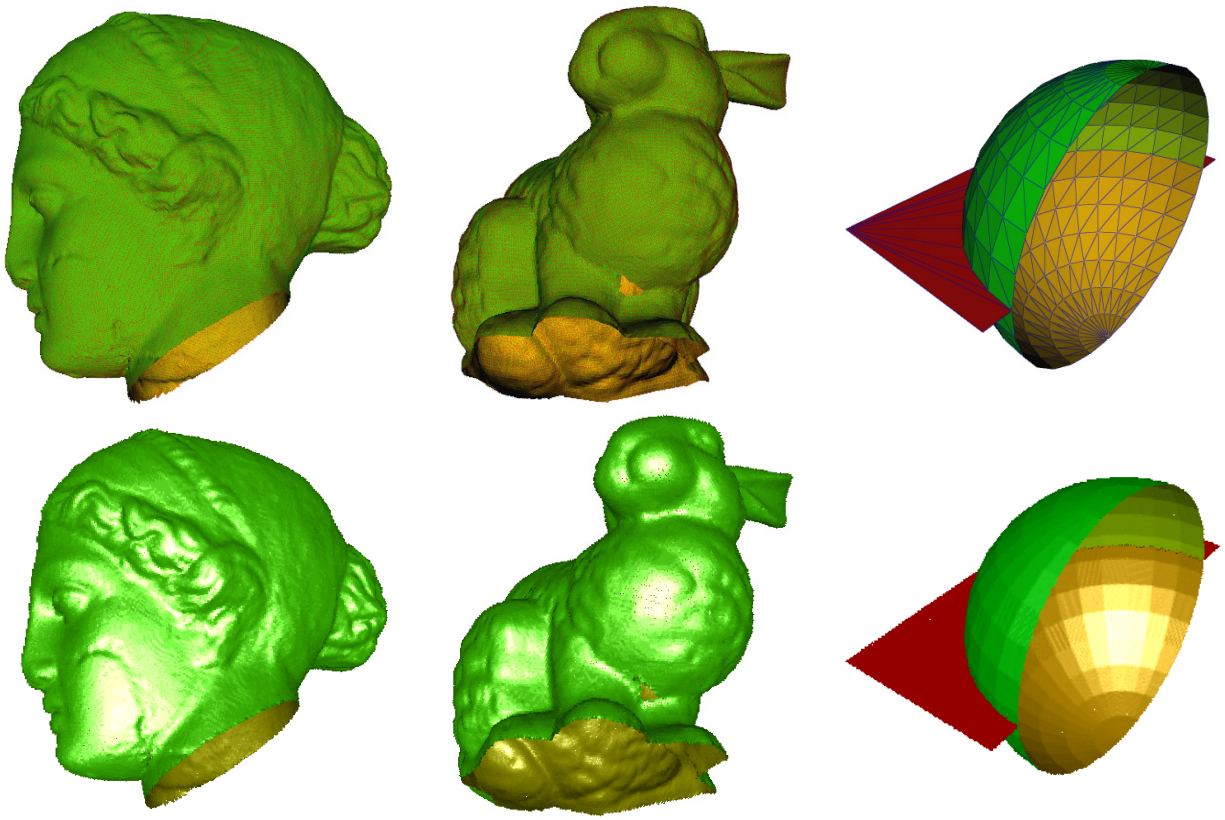

Figure 8: (Top) Input triangle meshes. (Bottom) Visualized surfaces represented in volume-sampled distance fields.

functions linearly without any correspondence.

6 Conclusions and future work

We have presented an interactive method of rendering non-manifold implicit surface taking advantage of hardware acceleration. Our proposed method of visualization can render implicit surfaces at sufficient quality and speed using commodity graphics hardware on a standard PC. In addition, our proposed representation of surface models is applicable to existing methods of geometric processing using implicit representations.

Implicit representations of surface models are powerful when they are deformed drastically and undergo topological changes which is difficult to be managed in a parametric representations. Free-form deformation (FFD) of the surface using implicit representation is one big challenge.

While graphics hardware has a great advantage especially in rendering, it also brings some disadvantages, for example, the limitation of data size and precision of rendering. Although the problem in size can be solved by applying adaptive sampling of the distance field, it is difficult to implement it in currently available graphics hardware, and still remains as future work.

7 Acknowledgement

We thank Akitake Makinouchi and Yoshinori Teshima for their comments and guidance. We are indebted to RIKEN and Japan Science and Technology Corporation for their

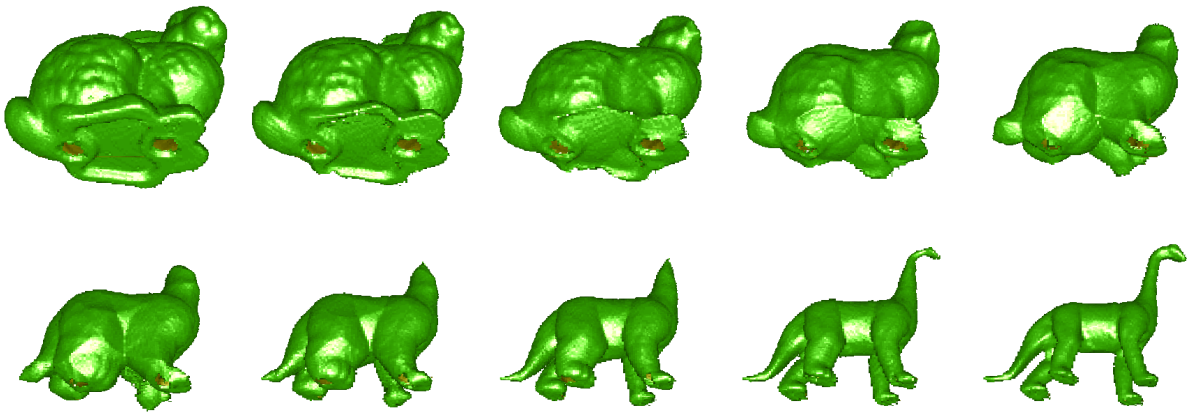

Figure 9: Example of blending two surface models by simple linear interpolation.

support of this research.

References

[1] Jules Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, Inc., 1997.

[2] Jules Bloomenthal and Keith Ferguson. Polygonization of non-manifold implicit surfaces. In *Proc. SIGGRAPH*, pages 309–316, 1995.

[3] K. S. Bonnell, D. R. Schikore, K. I. Joy, M. Duchaineau, and B. Hamann. Constructing material interfaces from data sets with volume-fraction information. In *Proc. Visualization 2000*, pages 367–372, 2000.

[4] Martin L. Brady, Kenneth Jung, HT Nguyen, and Thinh Nguyen. Two-phase perspective ray casting for interactive volume navigation. In *Visualization 97*, 1997.

[5] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proc. Eurographics / SIGGRAPH Workshop on Graphics Hardware*, pages 9–16, 2001.

[6] S.F. Frisken, R.N. Perry, A.P. Rockwood, and T.R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Proc. SIGGRAPH2000*, pages 249–254. ACM, July 2000.

[7] Dae Hyun Kim, Ulf Doering, and Beat Bruderlin. Polygonization of non-manifolds with the aid of interval operators. In *Proc. Implicit Surfaces*, pages 145–151, 2000.

[8] Leif Kobbelt, Mario Botsch, Ulrich Schwanecke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *Proc. SIGGRAPH*, 2001.

[9] M. Muuss and L. Butler. *Computer Graphics Techniques: Theory and Practice*, chapter Combinatorial Solid Geometry, B-Reps and n-Manifold Geometry. Springer Verlag, 1990.

[10] A. Paoluzzi, F. Bernardini, C. Cattani, and V. Ferrucci. Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics*, 12, 1993.

[11] Ronald N. Perry and Sarah F. Frisken. Kizamu: a system for sculpting digital characters. In *Proc. SIGGRAPH*, pages 47–56, 2001.

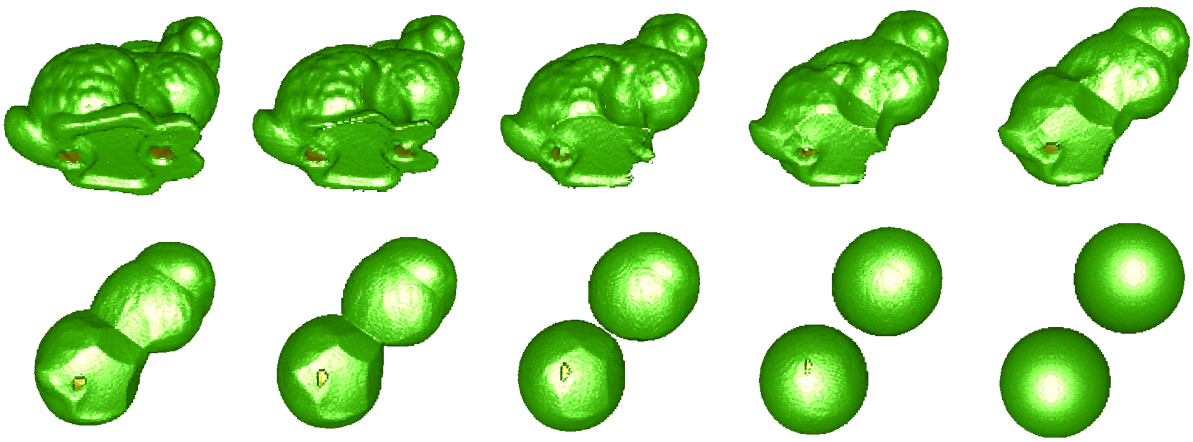

Figure 10: Example of blending two surface models with different topology

[12] H. Tuy and L. Tuy. Direct 2D display of 3D objects. *IEEE Mag. Computer Graphics and Applications*, 1984.

[13] A. P. Witkin and P. S. Heckbert. Using particles to sample and control implicit surfaces. In *Proc. SIGGRAPH*, pages 269–278, 1994.

[14] Shuntaro Yamazaki, Kiwamu Kase, and Katsushi Ikeuchi. Non-manifold implicit surfaces based on discontinuous implicitization and polygonization. In *Geometric Modeling and Processing*, pages 138–146. IEEE, July 2002.