# Shrink-Wrapped Boundary Face Algorithm for Mesh Reconstruction from Unorganized Points

Bon Ki Koo, Young Kyu Choi, Chang Woo Chu, Jae Chul Kim, and Byoung Tae Choi

*ABSTRACT—A new mesh reconstruction scheme for approximating a surface from a set of unorganized 3D points is proposed. The proposed method, called a shrink-wrapped boundary face (SWBF) algorithm, produces the final surface by iteratively shrinking the initial mesh generated from the definition of the boundary faces. SWBF surmounts the genus-0 spherical topology restriction of previous shrink-wrapping-based mesh generation techniques and can be applied to any type of surface topology. Furthermore, SWBF is significantly faster than a related algorithm of Jeong and others, as SWBF requires only a local nearest-point-search in the shrinking process. Our experiments show that SWBF is very robust and efficient for surface reconstruction from an unorganized point cloud.*

*Keywords—Shrink-wrapping, surface reconstruction, unorganized 3D points.*

## I. Introduction

In the past few decades, considerable studies have been conducted on the photo-realistic shape reconstruction of real objects. Shape reconstruction algorithms have two main stages: *data acquisition* and *surface reconstruction*. The data acquisition stage acquires an accurate, possibly unorganized, 3D point cloud from a real object, while the surface reconstruction stage converts the point cloud into a smooth surface.

The 3D point cloud itself is not adequate for other visualization operations such as volume calculation. Hence, there is a large literature on surface reconstruction from a point cloud. According to Curless and others [1], solutions to this problem have proceeded along two basic directions: *reconstruction from unorganized points* and *reconstruction that preserves the underlying structure of the acquired data.* The problem of surface reconstruction from unorganized points is known to be hard to solve since there is no prior assumption about the *connectivity* of the acquired 3D points, such as contour or range image. This letter focuses on the surface reconstruction from unorganized points.

A typical solution provided by Hoppe and others [2] introduces a *tangent plane* and a *signed distance function,* and produces a triangular mesh by a *volume-based reconstruction scheme*. But this method is not always robust in regions of high curvature or in the presence of systematic range distortions and outliers.

In this letter, we propose a new solution for mesh generation from an unorganized 3D point cloud based on an iterative relaxation scheme, called a shrink-wrapping process. The letter is organized as follows. Previous research work on surface generation based on shrink-wrapping is briefly described in section II. Our shrink-wrapped boundary face algorithm is described in section III. Experimental results are given in section IV, and section V concludes this letter.

## II. Previous Work on Surface Generation

Originally, the *shrink-wrapping*-based mesh generation technique was proposed by Kobbelt and others. [3]. They introduced a deformable surface scheme for converting a given unstructured triangle mesh into one having *subdivision connectivity* based on a simulation of the shrink wrapping process.

Recently, Jeong and others [4] extended the shrink-wrapping concept to produce a mesh model from unorganized points. For a given 3D point cloud, they make a bounding cube and linearly subdivide the six faces to get an initial cube-shaped mesh. They

Manuscript received Aug. 30, 2004; revised Jan. 11, 2005.

Bon Ki Koo (phone: +82 42 860 5033, email: bkkoo@etri.re.kr), Chang Woo Chu (email: cwchu@etri.re.kr), Jae Chul Kim (email: jaechul@etri.re.kr), and Byoung Tae Choi (email: btchoi@etri.re.kr) are with Digital Content Research Division, ETRI, Daejeon, Korea.

Young Kyu Choi (email: ykchoi@kut.ac.kr) is with the School of Information Technology, Korea University of Technology and Education, Cheonan, Korea.

repeatedly apply the *projection (shrinking)* and *smoothing* operations to metamorphose the initial mesh into one similar to the surface of the original object. Since the initial mesh is always a box shape, they have to restrict the topology of the reconstructed mesh: a *genus-0 spherical topology* that does not contain any holes in the object's surface. Therefore, their method cannot be applied to objects embedding holes such as a ring. Furthermore, for each vertex of the initial mesh, their method has to find the nearest point from all of the input points during the shrinking process, which is very time consuming.

In this letter, we also adopt the shrink-wrapping process to produce a mesh from unorganized points as in the method of Jeong and others [4]. The major difference in our method, named the shrink-wrapped boundary face (SWBF) algorithm, is the shape of the initial mesh. Instead of using the six faces of a bounding box, cell boundary faces defined in the next section are used as the initial mesh. This idea enables our method to overcome the topological restriction of the previous methods and considerably improves the computational time efficiency of generating a surface mesh.

## III. Shrink-Wrapped Boundary Face

Let $O_{real}$ be a real world object and $P=\{p_1, p_2, \cdots, p_n\}$ be point cloud sampled from $O_{real}$. No connectivity information between points $p_i, i=1,2,\cdots,n,$ is given. In this case, a surface reconstruction algorithm is a process that produces a polygonal mesh $M^P$ from $P$ such that $M^P \approx O_{real}$. SWBF generating $M^P$ from an unorganized point cloud $P$ includes the following four major stages: (i) partitioning the 3D space into a *cell space*, (ii) defining a *boundary face* and generating an *initial mesh*, (iii) *shrinking*, and (iv) surface *smoothing*. Figure 1 shows the conceptual illustrations of our definitions, and a detailed description follows in the next section.

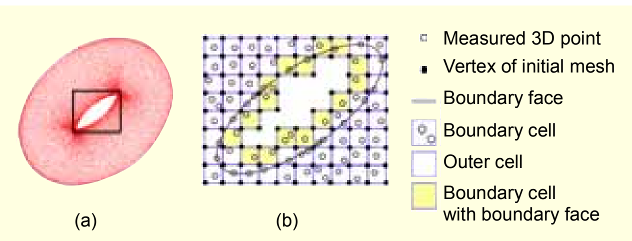



Fig. 1. Conceptual illustrations of our definitions: (a) measured 3D point cloud and (b) illustration of the terms in the 2D view of the rectangular area of (a).

### 1. Partition of the 3D Space into a Set of Cubes

First, we estimate a bounding box $B_P$ of $P$. Since the outside of $B_P$ may not produce any surface patch, we are concerned only with the inside of $B_P$. From the bounding box of $P$, we can define a cell and cell space as follows.

**Definition 1.** The *cell space* of a cloud of unorganized 3D points, $P$, is defined to be a dissection of $B_P$ by three orthogonal sets of equally spaced parallel planes and is denoted by $C_P$. Each cube, a component of the cell space, is called a *cell* and is denoted by $c$ or $c(i, j, k)$.

The size of each cell, or the resolution of the cell space, should be carefully selected according to the density of $P$. With the assumption that the density of the point sampling for each unit surface area is the same (or at least nearly similar), the major factor for selecting an adequate resolution could be the density of the point sampling. If the resolution is too low (compared with the point sampling density), the resulting mesh becomes coarse and cannot represent the detailed topology of the object. If it is too high, there may some unwanted holes in the resulting surface mesh. But its precise value is not very crucial compared with the previous method of [4].

### 2. Generating the Initial Mesh

Assume that a cell space $C_P$ is given from the original unorganized points $P$ with a certain resolution. The cells in $C_P$ can be divided into two groups as follows.

**Definition 2.** The *boundary cell*, denoted as $c_b$, is defined to be a cell containing one or more physical 3D points of $P$. Otherwise, the cell is defined to be an *outer cell* and is denoted as $c_o$.

Since $P$ is acquired only from the surface of $O_{real}$, we cannot classify whether a cell is inside or outside of the object. We can only differentiate a *boundary cell* from an *outer cell* contained inside or outside of the object. Boundary cell $c_b$ will have a high possibility of embedding the real surface, $O_{real}$.

The neighbors of cell $c$ are assumed to be cells which are adjacent to one of the six faces of $c$ and are denoted as $n(c)$. Udupa and others [5] used the term *O(1)-adjacency* for this kind of neighbor definition. We can define the boundary face as follows.

**Definition 3.** For boundary cell $c_b$, six faces exist in the directions of the O(1)-adjacent neighbor cells. A face of $c_b$ is defined to be a *boundary face* $f_b$ if the adjacent cell is an outer one. Otherwise, the face is defined to be an *inner face* and is denoted as $f_i$.

A boundary face can be interpreted as a simple approximation of the actual object surface, which is contained in the corresponding boundary cell. Consequently, the initial mesh can be defined from the boundary faces as follows.

**Definition 4.** The *initial mesh* denoted as $M^I$ is defined to be the set of boundary faces as

$$M^I = \{\forall f_b(c_b) \mid c_b \in C_P\} , \quad (1)$$

where $c_b$ represents the boundary cell in $C_P$.

The initial mesh $M^I$ in our method is a crude approximation of the surface of $P$ (or $O_{real}$) for further processing, but the important thing is that it usually preserves the topology of the original object unless the hole in object surface is too small compared with the resolution of the cell space. The crude surface is then iteratively metamorphosed into the original surface by shrinking and smoothing operations.

### 3. Shrinking

A shrinking step is applying an attracting force to each vertex—that is, a vector between the initial mesh $M^I$ and the given point cloud $P$. For a vertex $q_i$ of $M^I$, there is a boundary cell $c_b$ containing it. We search for the nearest 3D point in $P$ minimizing the Euclidean distance from $q_i$ to that point. Unlike in the method of [4] requiring a global search for optimal results, shrinking in SWBF can be done in a local manner to obtain a similar result. Since the nearest point $p_{near}$ should always be inside of $c_b$ or inside of the *O(3)-adjacent* neighbors of $c_b$, it is sufficient to search for only the 3D points inside a total of 27 cells for the nearest one from a mesh vertex. This greatly saves the processing time for the shrinking process. After finding $p_{near}$, the attracting force pushes $q_i$ toward $p_{near}$ as

$$q_i \leftarrow q_i + \alpha\{p_{near} - q_i\} . \quad (2)$$

The weight $\alpha$ (0.0 to 1.0) determines the amount of the attracting force. Since there is a possibility of sharing the same point by more than two mesh vertices, the full attraction force ($\alpha = 1$) may cause a non-manifold region in the surface. To avoid this, we provided a weight of less than 1.0 and experimentally chose $\alpha$ to be 0.5.

**Theorem 1.** The shrinking step previously described transforms the initial mesh $M^I$ in time proportional to the number of points in the original unorganized point cloud $P$.

*Proof.* As previously described, it is sufficient to consider only the 3D points inside a total of 27 cells when searching for the nearest one from a mesh vertex $q_i$ of $M^I$. From the view point of a 3D point $p$ in $P$, the distance calculation from $p$ to the mesh vertices can be done only for those vertices embraced in the 27 cells: the cell $c_b$ containing $p$ and 26 O(3)-adjacent neighbors of $c_b$. There are eight supporting vertices in each cell, and some of them participate in the initial mesh $M^I$ as vertices. Thus, a point $p$ in $P$ should be considered when searching for the nearest vertex from at most 216 (=27 × 8) mesh vertices. Consequently, the time complexity for the shrinking step is $O(216n)=O(n)$, where $n$ is the total number of points in the original unorganized point cloud $P$.

Compared with the method of [4], our method is much faster since it requires $O(mn)$ time for a shrinking step, where $m$ is the total number of mesh vertices in $M^I$. Since $m$ should increase nearly proportionally to $n$ in physical experiments, the actual time complexity of [4] becomes $O(n^2)$ for a shrinking step.

### 4. Surface Smoothing

The smoothing step tries to relax the shrink-wrapped surface for achieving a uniform vertex sampling, and we have adopted the same method used in [4]. We employ the approximation of Laplacian $\mathcal{L}$ as in [6]. This is the average vector of 1-neighbor edge vectors of a given vertex, and thus a surface shrinkage effect may occur. Therefore, we take only the tangential component of $\mathcal{L}$, which is perpendicular to the vertex normal. The processing time of the smoothing step is proportional to the number of vertices in the initial mesh $M^I$ since for a mesh vertex $q_i$ of $M^I$ there are at most six neighboring vertices, $O(6m) = O(m)$.

## IV. Experiments

Figure 2 illustrates the difference between SWBF and the method proposed by Jeong and others [4] for a synthetic ring data containing a hole in the surface. When embedding any kind of topology, the initial mesh in SWBF is the key for resolving the genus-0 spherical topology restriction of previous shrink-wrapping-based reconstruction methods, which can be seen by the comparable results in Figs. 2(d) and 2(g). Notice

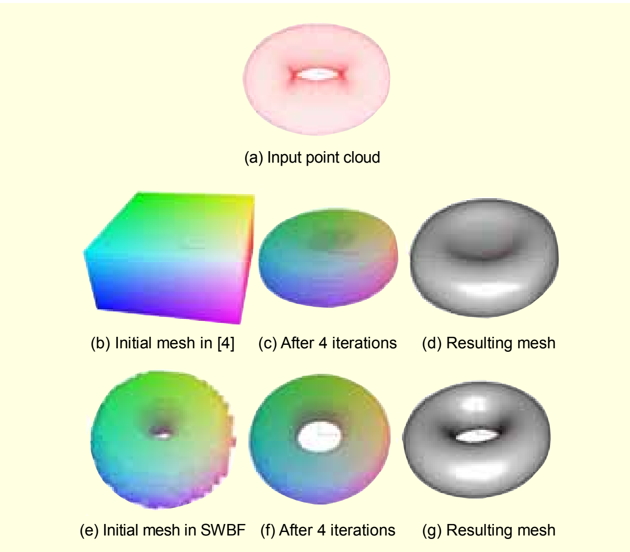
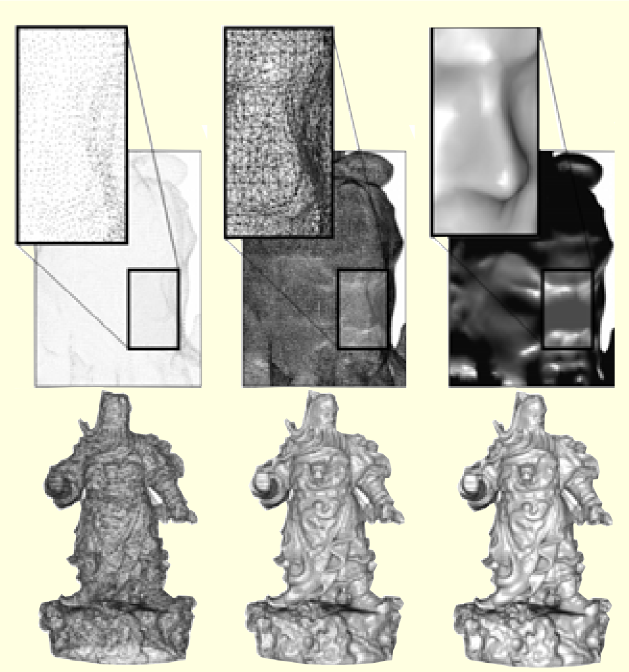



Fig. 2. Comparison with [4] (cell resolution=21): (a) input data, (b) - (d) results from [4], and (e) - (g) results by SWBF.



(a) Input point cloud (b) Result (mesh) (c) Result (shading)

Fig. 3. Reconstruction result for the *General* data: (a) input point cloud, (b) shrink-wrapped surface (3 iterations), and (c) reconstructed surface in Gouraud shading.

Table 1 Reconstruction summary for the *General* data.

| | | |
|---|---|---|
| Input point cloud | Number of 3D points | 1,658,574 |
| Cell space | Maximum resolution | Y:200 |
| | Number of boundary cells | 96,164 |
| | Average number of points in a boundary cell | 17.2473 |
| Reconstructed mesh | Number of vertices | 198,267 |
| | Number of triangular faces | 398,784 |

that the initial mesh in SWBF, Fig. 2(e), also envelops all of the original 3D points in Fig. 2(a). As shown in the resulting mesh in Fig. 2(g), four iterations of the shrink-wrapping process would be sufficient in SWBF for general cases. But as Fig. 2(d) shows it is still not enough in [4] because the shape of the initial mesh, shown in Fig. 2(b), is far from the real object.

Figure 3 shows the reconstruction results for the *General* data. As you can see in Fig. 3, SWBF works well for generating a mesh even with only three occurrences of the shrinking and smoothing process. Table 1 summarizes the reconstruction results. The overall processing time was less than 60 seconds on a Pentium 2.0 MHz PC. Since SWBF searches only O(3)-adjacent neighbors in finding the nearest points, it is much faster than previous works [2], [4]. In our experiment, even one iteration of the shrinking process based on a global search, as in [4] for the *General* data, takes more than several hours.

## V. Conclusion

In this letter, we introduced a novel mesh reconstruction algorithm from an unorganized point cloud. SWBF overcomes the genus-0 spherical topology restriction of the previous shrink-wrapping based mesh generation methods by using the boundary faces as the initial mesh. Furthermore, it is much faster since it requires only a local nearest-point-search in the shrinking process. According to experiments, SWBF is found to be very robust and efficient, and is expected to be a general solution for reconstructing a mesh from an unorganized point cloud.

## References


[1] B. Curless and M. Levoy, “A Volumetric Method for Building Complex Models from Range Images,” *SIGGRAPH '96*, Aug. 1996, pp. 303-312.

[2] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface Reconstruction from Unorganized Points,” *SIGGRAPH '92*, Jul. 1992, pp. 71-78.

[3] L. Kobbelt, J. Vorsatz, U. Labsik, and H. Seidel. “A Shrink Wrapping Approach to Remeshing Polygonal Surfaces,” *Proc. Eurographics '99*, vol. 18, no. 3, Sept. 1999, pp. 119-129.

[4] W. Jeong and C. Kim, “Direct Reconstruction of Displaced Subdivision Surface from Unorganized Points,” *Graphical Models*, vol. 64, issue 2, Mar. 2002, pp. 78-93.

[5] J. Udupa, S. Srihari, and G. T. Herman, “Boundary Detection in Multidimensions,” *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol. 4, no. 1, 1982, pp. 41-50.

[6] G. Taubin, “A Signal Processing Approach to Fair Surface Design,” *SIGGRAPH '95*, Aug. 1995, pp. 351-358.