ELSEVIER

Computer-Aided Design 36 (2004) 1–9

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

A mesh reconstruction algorithm driven by an intrinsic property of a point cloud

Hong-Wei Lin[a], Chiew-Lan Tai[b], Guo-Jin Wang[a,*]

[a]Department of Mathematics, State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, People's Republic of China
[b]Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, People's Republic of China

Received 19 October 2002; received in revised form 7 March 2003; accepted 21 March 2003

Abstract

This paper presents an algorithm for reconstructing a triangle mesh surface from a given point cloud. Starting with a seed triangle, the algorithm grows a partially reconstructed triangle mesh by selecting a new point based on an intrinsic property of the point cloud, namely, the sampling uniformity degree. The reconstructed mesh is essentially an approximate minimum-weight triangulation to the point cloud constrained to be on a two-dimensional manifold. Thus, the reconstructed surface has only small topological difference from the surface of the sampled object. Topological correct reconstruction can be guaranteed by adding a post-processing step.
© 2003 Elsevier Ltd. All rights reserved.

Keywords: Point cloud; Intrinsic property; Reconstruction

1. Introduction

Given a point cloud sampled from the surface of a three-dimensional object, the objective of the reconstruction problem is to recover the geometric shape from the point cloud. The problem has found many important applications in reverse engineering, virtual reality and computer vision. To solve the problem, a reconstructed mesh approximating or interpolating the point cloud must be computed. Many mesh reconstruction methods have been proposed; comprehensive surveys of these methods can be found in Ref. [1,2]. The methods fall into three categories.

In *sculpting-based approaches*, the Delaunay triangulation of the point cloud is first constructed, followed by the extraction of triangles or triangular patches representing the object shape. For example, Boissonnat's method [3] applied the 3D Delaunay triangulation to obtain the convex hull of the point cloud. If not all the points in the point cloud are on the boundary of the convex hull, then some tetrahedron in the triangulation must be deleted in turns until all points lie on the boundary P of the polyhedral shape so obtained, while ensuring that P remains a polyhedron. Other algorithms in this category are *alpha-shape* [4–6], *γ-graphs* [7], *β-skeleton* [8], *crust* algorithm [9–10], and *UmbrellaFilter* algorithm [11].

In *contour-tracing approaches*, for which Hoppe's [12] is a typical example, the triangle mesh is obtained by contouring the zero set of a signed distance function determined by the point cloud. The contour-tracing approaches, which produce approximating rather than interpolating surfaces, inherently do some low-pass filtering of the point cloud. This is desirable in the presence of noise, but causes some loss of information [9]. Other examples of methods in this category are presented in Ref. [13–15].

Region-growing approaches construct the mesh starting with a seed triangle patch, and progressively adding new triangles attached to the partially constructed mesh. The boundary edges are considered active edges, to which new triangles are added. The key problem of approaches in this category is how to select a point to form a new triangle with an active edge. In the *BPA* algorithm [16], a ball with user-specified radius pivots around an active edge until it touches another point in the point cloud; the point being touched is selected. Huang and Menq [17] projected the k nearest points of each endpoint of an active edge, respectively, onto the plane defined by the triangle adjacent to the active edge.

* Corresponding author.
E-mail address: wgj@math.zju.edu.cn (G.-J. Wang).

0010-4485/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0010-4485(03)00064-2

A point is chosen among the k points based on the minimal length criterion to form a triangle with the active edge. Petitjean and Boyer presented another method based on regular interpolation [18].

To ensure the correct post-processing of the reconstructed mesh surface in CAD applications, the topology of the reconstructed mesh surface has to be correct, that is, the reconstructed mesh surface is homeomorphic to the surface of the sampled object. Amenta et al. [9] presented an algorithm that guarantees topological correctness for point clouds satisfying the condition of γ-sample. Petitjean and Boyer [18] proposed a condition based on regular set, which is easier to verify. If the point cloud is a regular set, their interpolation algorithm guarantees topological correctness. The solution of Adamy, Giesen and John [11] involves two stages: they first reconstructed the triangle mesh using the *UmbrellaFilter* algorithm, and then achieve topology correctness by solving a linear programming (LP) problem. However, if the topology of the reconstructed surface differs too much from that of the sampled surface, the computational complexity of the LP problem is very large, and its solution may not exist [11].

In this paper, we present an algorithm called *Intrinsic Property Driven (IPD) algorithm,* which falls under the category of region-growing approaches. As stated above, the key problem is how to choose a point for forming a new triangle with an active edge. In the BPA algorithm [16], the point cloud must be scanned many times using balls of different radii to reconstruct the whole surface. Clearly, the surface reconstructed by the BPA algorithm relies on the user-specified ball radii. Huang and Menq's method [17] leads to two unreasonable results. First, the distances among the points are distorted after projection, so some errors may occur in the reconstructed mesh for the distances are only used to decide which point to choose right. Second, the reconstructed mesh surface is dependent on the user-specified parameter k nearest points. In short, the quality of the mesh surfaces reconstructed by the BPA algorithm or Huang and Menq's method relies on user-specified parameters. In order to overcome this limitation, we introduce the concept of sampling uniformity degree.

We define the sampling uniformity degree at a point as the ratio of the lengths of the longest edge and shortest edge incident to the point. It is clearly an intrinsic property of a point cloud. The IPD algorithm searches for a new point based on the sampling uniformity degree. Unlike algorithms requiring user-specified parameters, the mesh surface reconstructed by the IPD algorithm completely relies on this intrinsic property of the point cloud. Furthermore, we minimize the harmonic energy [19] in the reconstruction, so that the reconstructed mesh is essentially the minimum-weight triangulation to the point cloud restricted to be on the surface of the sampled object. Experimental results show that the difference in topology between the reconstructed surface and the surface of the sampled object is small. Thus, from the output of IPD algorithm, we can easily obtain the topologically correct surface using the post-processing method in Ref. [11].

This rest of the paper is organized as follows. In Section 2, we introduce some basic concepts and definitions for presenting the IPD algorithm. The IPD algorithm is discussed in detail in Section 3. Section 4 presents a new criterion for evaluating the quality of a reconstructed mesh surface in the topological sense. In Section 5, we present some experimental results. We conclude the paper in Section 6.

2. Basic concepts

Let **P** be an arbitrary point in a point cloud. The *sampling uniformity degree* at **P** is defined as the length ratio between the longest edge and the shortest edge incident to **P** (Fig. 1). It reveals the distribution of the sampling points near **P** and is an intrinsic property of the point cloud. Values close to 1 imply uniform sampling distribution, and larger values imply more non-uniformity in the sampling distribution. According to the definition, the exact sampling uniformity degree at a point **P** can be calculated only when all the edges incident to the point **P** are already reconstructed. Thus, in the IPD algorithm, we only approximate the sampling uniformity degree, by considering only the reconstructed edges incident to **P**.

In the mesh reconstruction procedure, each newly reconstructed edge is considered an *active edge*. To search for a new point to form a new triangle with an active edge $\mathbf{P}_i\mathbf{P}_j$, we must determine an *influence region* for each active edge, which may or may not contain any new point. This influence region is an extruded polyhedron containing the edge $\mathbf{P}_i\mathbf{P}_j$. Since the distance between the new point and the edge $\mathbf{P}_i\mathbf{P}_j$ is related to the distribution of points near $\mathbf{P}_i$ and $\mathbf{P}_j$, it is reasonable that the size of the influence region should be dependent on sampling uniformity

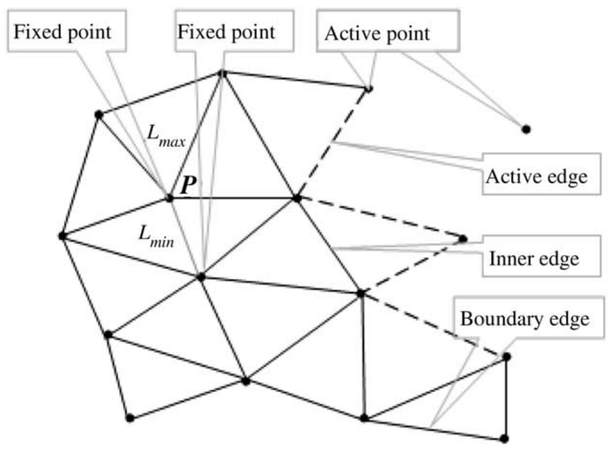

Fig. 1. Some basic concepts. Specifically, the *sampling uniformity degree* at **P** is the length ratio $L_{\max}/L_{\min}$.

degrees of points near $\mathbf{P}_i$ and $\mathbf{P}_j$. We shall explain in detail how we construct the influence region of an active edge in Section 3.2.

An edge is called an *inner edge*, if it has two adjacent faces (see Fig. 1). A point is called a *fixed point*, if all its incident edges are inner edges. Otherwise, if the influence region of an active edge contains no point except for fixed points, the edge is a *boundary edge* (i.e. it has only one face adjacent to it. See Fig. 1 and Fig. 3c). Thus, a point is called an *active point* if it has not been processed (that is, no edge is incident to it), or if there is an active edge or boundary edge incident to it.

3. IPD algorithm

Suppose the point cloud sampled from the surface of a three-dimensional object is sufficiently dense. We present the IPD algorithm for reconstructing a mesh surface from a given point cloud.

We assume that the bounding box of the point cloud is represented as a voxel set for efficient lookup of the neighboring points of a given point. All points in a voxel are organized into a linked list. To begin, we choose a seed triangle from the point cloud and add its three edges to the active-edge queue. In each iteration, we de-queue an edge and determine its influence region. If there exist some active points in its influence region, we choose a point from them for forming a new triangle (detail in Section 3.3), and then add the newly reconstructed edges to the active-edge queue. Otherwise, if there is no active point in the influence region, the algorithm concludes that it is a boundary edge. The procedure is repeated until the active-edge queue is empty. Fig. 2 shows the outline of the IPD algorithm. The parameters i, j and k denote the indices of points, $e_{i,j}$ denotes the edge constructed between the points i and j, *AEQ* denotes the active-edge queue.

```
IPD Algorithm (Point Cloud C)
1.  Search for Seed Triangle 0;
2.  while ($e_{ij}$ = Get Active Edge (AEQ));
3.          $p_k$ = Search for New Point ($e_{ij}$);
4.          if($p_k$ is an active Point in C)
5.                  Create New Triangle (i,j,k);
6.          else
7.                  $e_{ij}$ is a boundary edge;
8.  return;
```

Fig. 2. The *IPD* Algorithm.

3.1. Seed triangle selection

In the *IPD* algorithm, we first construct the seed triangle as follows:

1. Search for a point **P** whose z-coordinate is the largest in the point cloud;
2. Search for point **Q** that is nearest to **P** and form a line segment **L** between them;
3. Construct a cylinder with **L** as its axis, the midpoint of **L** as its center, the length of **L** as its height and diameter. Uniformly augment its radius and height (along opposite directions of its axis) until the cylinder contains some point in the point cloud.
4. Among the points within the cylinder, choose one point **R** so that the sum of the lengths of the two edges connecting **R** and the points **P** and **Q**, respectively, is the minimum. Let the seed triangle be $\Delta \boldsymbol{PQR}$.

We choose such a triangle as the seed for the convenience of adjusting its normal vector to be outward: if the inner product of the normal vector of the seed triangle and the vector (0,0,1) is positive, we have the desired normal vector; otherwise we reverse its direction. Once the outward normal vector of a triangle A is known, we can determine the direction of the normal vector of a newly generated triangle B, which is adjacent to A. Specifically, if the inner product of the two normal vectors is positive, then normal vector of B is also in the outward direction.

3.2. Influence region of edge

As stated in Section 2, the size of the influence region of an active edge $e_{i,j}$ connecting $\mathbf{P}_i$ and $\mathbf{P}_j$ should be dependent on the sampling uniformity degrees at the points near $\mathbf{P}_i$ and $\mathbf{P}_j$. In fact, in order to ensure that the influence region is suitably large, its size is determined from the product of the maximum sampling uniformity degree at vertices $\mathbf{P}_i$ and $\mathbf{P}_j$ and the average length of the minimum edges incident to $\mathbf{P}_i$ and $\mathbf{P}_j$, respectively. Let s be the result of this product, and let $\mathbf{P}_m$ be the midpoint of the edge $e_{i,j}$. For the triangle adjacent to $e_{i,j}$ (there is only one adjacent triangle since $e_{i,j}$ is active), let $\mathbf{P}_k$ be its third vertex, **P** be its barycenter, and **N** its normal vector, which is in the direction of $\mathbf{P}_k\mathbf{P}_i \times \mathbf{P}_k\mathbf{P}_j$ We calculate the influence region of the active edge $e_{i,j}$ as follows (see Fig. 3).

Fig. 3a shows the influence region of the active edge $e_{i,j}$. The dashed polygon is the projection of the influence region onto the plane defined by the triangle adjacent to $e_{i,j}$. Each boundary face B_i of the influence region is defined by a ($\mathbf{N}_i$,$\mathbf{P}_i$) pair, where $\mathbf{N}_i$ denotes the normal vector of the plane containing the face and $\mathbf{P}_i$ is a point on the plane. The top

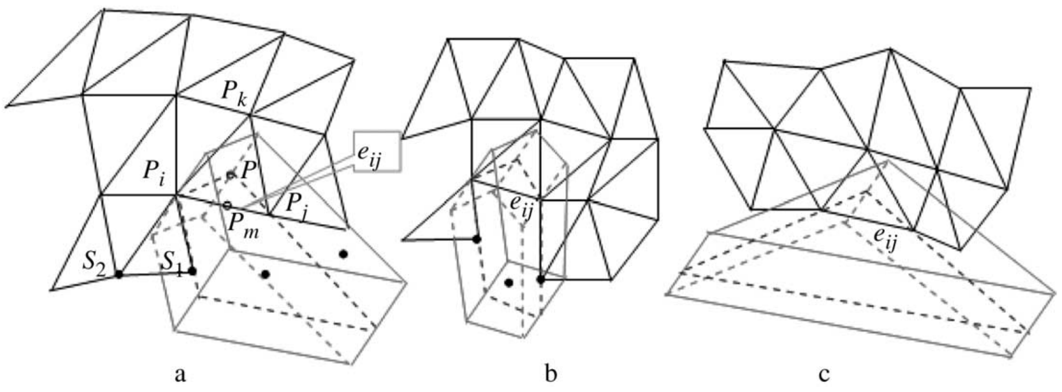

Fig. 3. The influence region of the edge $e_{i,j}$ (the red polyhedron), and the blue dashed polygon is its projection onto the plane defined by the triangle adjacent to $e_{i,j}$. (The black dots represent the sampling points.).

boundary face B_1 is defined by $\mathbf{N}_1 = \mathbf{N}$ and $\mathbf{P}_1 = \mathbf{P}_m + s\mathbf{N}$. The bottom boundary face B_2 is defined by $\mathbf{N}_2 = -\mathbf{N}$ and $\mathbf{P}_2 = \mathbf{P}_m - s\mathbf{N}$. For the boundary face B_3 containing edge $\mathbf{PP}_i$, $\mathbf{N}_3$ is the normalized vector of $\mathbf{N} \times \mathbf{PP}_i$, and $\mathbf{P}_3 = \mathbf{P}_i$. For the boundary face B_4 containing edge $\mathbf{PP}_j$, $\mathbf{N}_4$ is the normalized vector of $\mathbf{N} \times \mathbf{PP}_j$, and $\mathbf{P}_4 = \mathbf{P}_j$. The last boundary face B_5 has $\mathbf{N}_5$ in the direction of $\mathbf{P}_i\mathbf{P}_j \times \mathbf{N}$ and $\mathbf{P}_5 = \mathbf{P}_m + s\mathbf{N}_5$. In order to guarantee the geometry integrity of the reconstructed mesh, we need to ensure that the intersection between the newly generated triangular patch and the existing triangular patches is either empty or existing active or boundary edges. Thus, if there are edges incident to $\boldsymbol{P}_i$ (respectively, $\boldsymbol{P}_j$) lying on the same side of the face B_3 (resp. B_4) as the edge $\mathbf{P}_i\mathbf{P}_j$ (e.g. $\mathbf{P}_i\mathbf{S}_2$ and $\mathbf{P}_i\mathbf{S}_1$ in Fig. 3a), then we add a new boundary face B_6, where $\mathbf{P}_6 = \mathbf{P}_i$, $\mathbf{N}_6 = \mathbf{N} \times \mathbf{P}_i\mathbf{P}_{\text{left}}$, with the vector $\mathbf{P}_i\mathbf{P}_{\text{left}}$ chosen such that it forms the smallest angle with $\mathbf{P}_i\mathbf{P}_j$ among all edges lying on that side. For example, $\mathbf{P}_{\text{left}} = \mathbf{S}_1$ in Fig. 3a. We illustrate some examples of influence regions in Fig. 3a–c. However, the smallest angle criterion is insufficient in guaranteeing geometry integrity, and further test is performed as described in Section 3.3.

We note that Ref. [17] also introduced the concept of influence region of active edge, which was defined to be in the plane containing the triangular patch adjacent to the active edge. In c ontrast, we define the influence region of an active edge in the three-dimensional space and its size is determined by an intrinsic property of point clouds—the *sampling uniformity degree*.

3.3. New vertex search

In each iteration, if the influence region of an active edge $e_{i,j}$ contains some active points, the IPD algorithm chooses one of them to form a new triangle with $e_{i,j}$ (e.g. There are two active points in Fig. 3a, that is, $\boldsymbol{S}_1$ and another point). Two common criteria are the *minimal area criterion* [20] and the *minimal length criterion* [15]. We implemented and tested these two criteria and found that the reconstructed surface using the *minimal length criterion* is visually better than that using the *minimal area criterion*. While the reconstructed surfaces using the *minimal length criterion* often have large difference in topology with the surfaces of the sampled objects. Thus, we propose a new criterion called *weighted minimal length criterion* for selecting a new point.

Clearly, it is desirable to have the reconstructed surface as close as possible to the surface of the sampled object. For a given point cloud, suppose there are n different topologically correct reconstructed triangle meshes with straight edges. Each of these straight-edge triangle meshes corresponds to a curved-edge triangle mesh embedded onto the surface of the sampled object. Although the n curved-edge triangle meshes have different connections among the vertices, they represent the same surface. Hence, the reconstructed mesh surface that is closest possible to the sampled surface is the one with the minimum metric distortion from the corresponding curved-edge mesh surface. Examples of metrics are the aspect ratio of triangles, and the length of mesh edges. From Ref. [19], the mesh S' with minimum metric distortion to a mesh S minimizes the following functional:

$$E = \frac{1}{2}\sum k_{i,j}\|h(\mathbf{P}_i) - h(\mathbf{P}_j)\|^2. \tag{1}$$

where $k_{i,j}$ is a coefficient, $\mathbf{P}_i$ and $\mathbf{P}_j$ are the two vertices of an edge in S, $h(\mathbf{P}_i)$ and $h(\mathbf{P}_j)$ are the two vertices of the corresponding edge in S', and $h(\cdot)$ is called a harmonic map. Let the length of the edge $e_{i,j}$ in a curved-edge triangle mesh be $L_{i,j}$, and the area of a curved-edge triangle patch $\{i, j, k\}$ be $A_{i,j,k}$.

In Ref. [19], the connections among the vertices are known, and the map $h(\cdot)$ is unknown. In our case, the map $h(\cdot)$ is an identity map, and the connections among vertices are unknown. Therefore, the problem becomes that of solving the *minimum-weight triangulation* restricted to lie on the surface of the sampled object. However, solving the minimum-weight triangulation problem is very hard even for planar point sets [21–22]. Hence, for simplification, we adopt a heuristic strategy to solve the approximate minimum-weight triangulation, restricted to be on the sampled surface. That is, suppose the triangle patch $\{i, j, k\}$ is adjacent to the active edge $e_{i,j}$, we select the new vertex $\mathbf{P}_m$ for $e_{i,j}$ such that the triangle patch $\{i, j, m\}$ minimizes the following sum:

$$k_{i,j}\|\mathbf{P}_i - \mathbf{P}_j\|^2 + k_{i,m}\|\mathbf{P}_i - \mathbf{P}_m\|^2 + k_{j,m}\|\mathbf{P}_j - \mathbf{P}_m\|^2 \tag{2}$$

Since $e_{i,j}$ is adjacent to two triangle patches $\{i, j, k\}$ and $\{i, j, m\}$, we calculate the coefficient $k_{i,j}$ as follows [19]:

$$k_{i,j} = (L_{i,k}^2 + L_{j,k}^2 - L_{i,j}^2)/A_{i,j,k} + (L_{i,m}^2 + L_{j,m}^2 - L_{i,j}^2)/A_{i,j,m}. \tag{3}$$

For the newly generated edges $e_{i,m}$ and $e_{j,m}$, which possess only one adjacent triangle patch, respectively, we approximate $k_{i,m}$ and $k_{j,m}$ according to the following equation:

$$k_{i,m} = k_{j,m} = 2 \times (L_{i,m}^2 + L_{j,m}^2 - L_{i,j}^2)/A_{i,j,m}. \tag{4}$$

As stated above, if the influence region of an active edge $e_{i,j}$ contains some active points, the IPD algorithm firstly chooses the point that minimizes Eq. (2) from them, and then performs the following test for geometry integrity on the chosen point. It checks whether the intersection between the newly generated triangular patch (defined by the chosen point and edge $e_{i,j}$) and those existing triangular patches adjacent to the points in the influence region of the edge $e_{i,j}$ is empty or an existing active or boundary edge. If so, the choice of point is confirmed. Otherwise, the IPD algorithm chooses another point that minimizes Eq. (2) from the remaining active points and performs the same geometry integrity test. If all active points have been processed and none satisfies the test for geometry integrity, IPD algorithm considers the edge $e_{i,j}$ as a boundary edge.

3.4. New triangle construction

After selecting an active point $\mathbf{P}_m$, the *IPD* algorithm proceeds to construct a new triangle $\Delta \boldsymbol{P}_i\boldsymbol{P}_j\boldsymbol{P}_m$ as follows:

1. Generate $\Delta\mathbf{P}_i\mathbf{P}_j\mathbf{P}_m$
    1.1. If the edge $\mathbf{P}_i\mathbf{P}_m$ does not exist, construct the edge $\mathbf{P}_i\mathbf{P}_m$, and add it to the active edge queue.
    1.2. If the edge $\mathbf{P}_j\mathbf{P}_m$ does not exist, construct the edge $\mathbf{P}_j\mathbf{P}_m$, and add it to the active edge queue.
2. Classify the edge $\mathbf{P}_i\mathbf{P}_j$ as an inner edge, and classify each of the edges $\mathbf{P}_i\mathbf{P}_m$ and $\mathbf{P}_j\mathbf{P}_m$ as an inner edge if it has two adjacent faces, or as an active edge otherwise.
3. Classify the point $\mathbf{P}_i$, $\mathbf{P}_j$ and $\mathbf{P}_m$ as fixed point or active point, respectively.

4. Criterion for evaluating surface topological quality

If the surface of the sampled object is closed, and it is homeomorphic to the reconstructed mesh surface, then the number of the triangle patches (denoted t) and the number of vertices (denoted v) in the mesh surface must satisfy the following Euler formula [11]:

$$t = 2 \times v + 4 \times (g - 1) \tag{5}$$

where g is the genus of the sampled object. Most reconstruction algorithms do not guarantee that the reconstructed mesh surface is homeomorphic to the sampled surface, and thus do not always satisfy formula (5). The formula is a necessary condition for the homeomorphism, and we can also derive the following measurement e, which reveals the topological difference between the two surfaces:

$$e = |t - (2 \times v + 4 \times (g - 1))| \tag{6}$$

where $|\cdot|$ denotes absolute value. Thus, it can be used as a criterion for evaluating the quality of the reconstructed mesh surface in the topological sense.

Notably, the above formulae are applicable only to closed surfaces. For objects with boundary, such as the U-shape in Fig. 4, the formula must be modified. Suppose that the sampled object has n open boundaries, then the reconstructed mesh surface should have n corresponding boundary loops. Suppose these boundary loops contain a total of m boundary edges. To produce a closed reconstructed surface, we introduce a vertex to each boundary loop, and use each boundary edge in the boundary loop to form a triangle with the new vertex (see Fig. 4); that is, we transform the mesh surface into a closed object by adding m triangles. The closed object now has $(t + m)$ triangles and $(v + n)$ vertices. Hence, if the reconstructed mesh surface with boundaries is topologically correct, it must satisfy the following formula.

$$t + m = 2 \times (v + n) + 4 \times (g - 1), \tag{7}$$

where g is the genus of the sampled object after capping all boundaries. Accordingly, the difference formula can be modified to

$$e = |(t + m) - (2 \times (v + n) + 4 \times (g - 1))|. \tag{8}$$

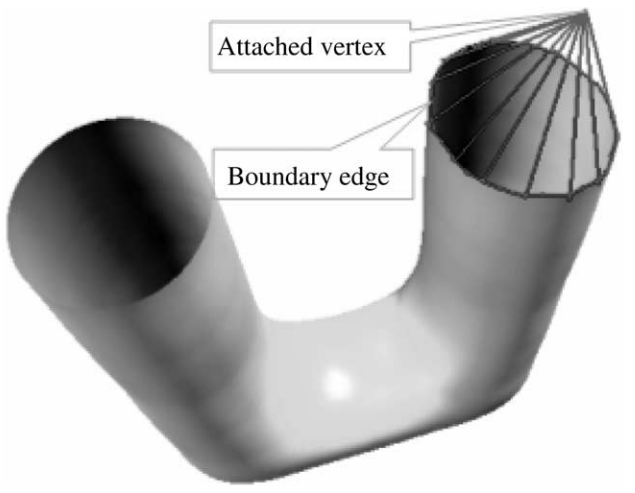

Fig. 4. Attaching a vertex to close up a boundary.

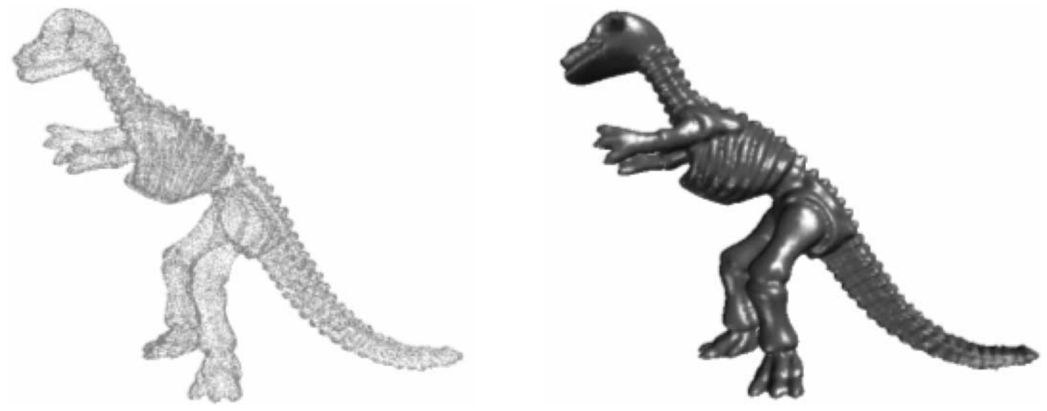
Fig. 5. Dragon. Left: point cloud. Right: reconstructed mesh surface.

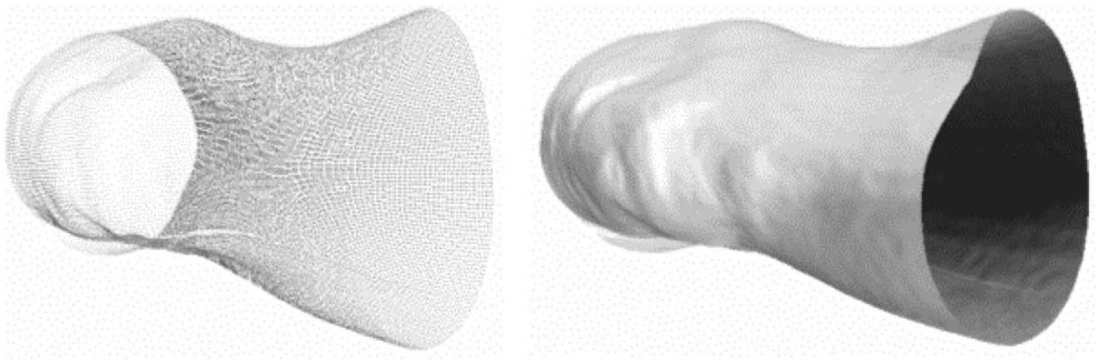
Fig. 6. Knee. Left: point cloud. Right: reconstructed mesh surface.

Like other algorithms, the *IPD* algorithm cannot guarantee the homeomorphism between the reconstructed surface and the surface of the sampled object. However, experimental results presented in the Section 5 show that mesh surfaces reconstructed by the *IPD* algorithm have small topological difference from the sampled surfaces. Consequently, topologically correct reconstructed surfaces can be easily obtained using a post-processing method such as that in Ref. [11].

5. Results

Figs. 5–11 show some mesh surfaces reconstructed using the *IPD* algorithm. The dragon with closed surface is reconstructed from dense point cloud. The knee is an example with boundaries, and it is reconstructed from a point cloud with scan line distribution. The cube with holes is a topologically complex model of genus 5 and is reconstructed from sparse point cloud. The Fan-disk,

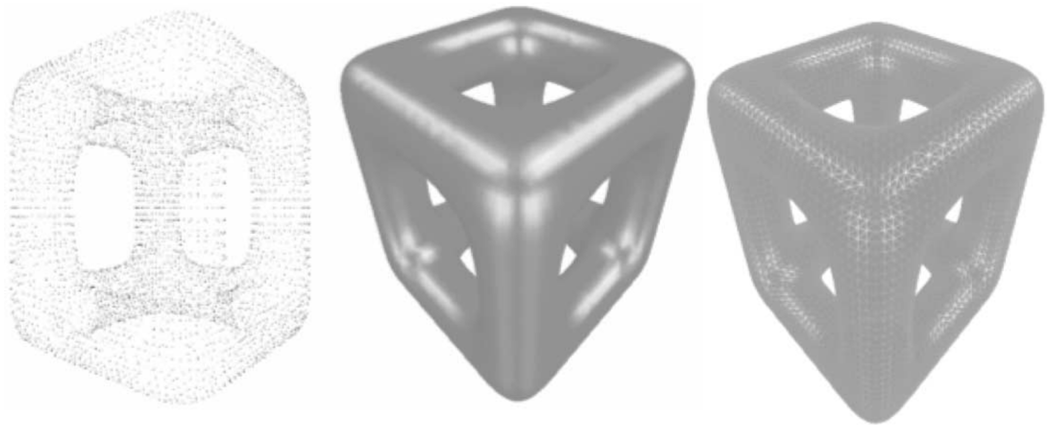
Fig. 7. Cube with holes. Left: point cloud. Middle: reconstructed rendered surface. Right: reconstructed mesh.

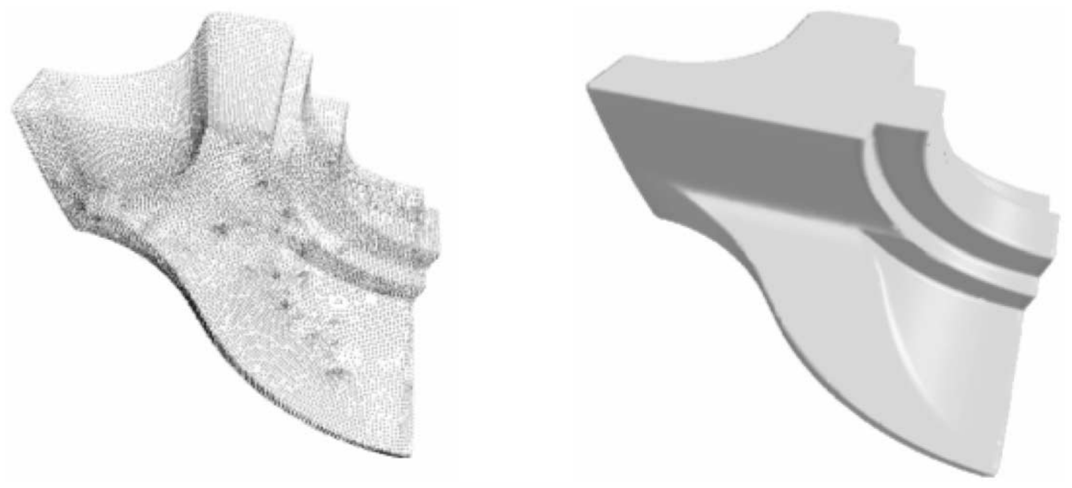
Fig. 8. Fan-disk. Left: point cloud. Right: reconstructed mesh surface.

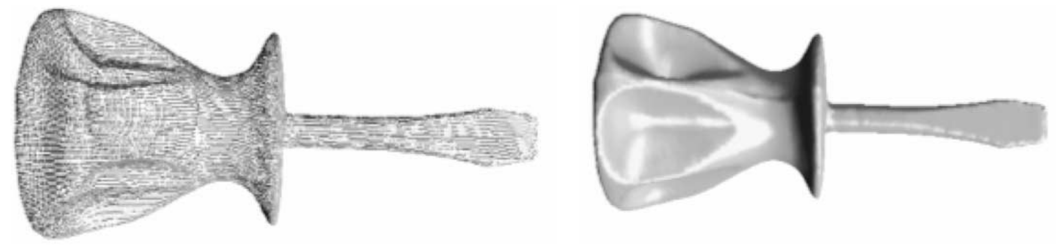
Fig. 9. Screwdriver. Left: point cloud. Right: reconstructed mesh surface.

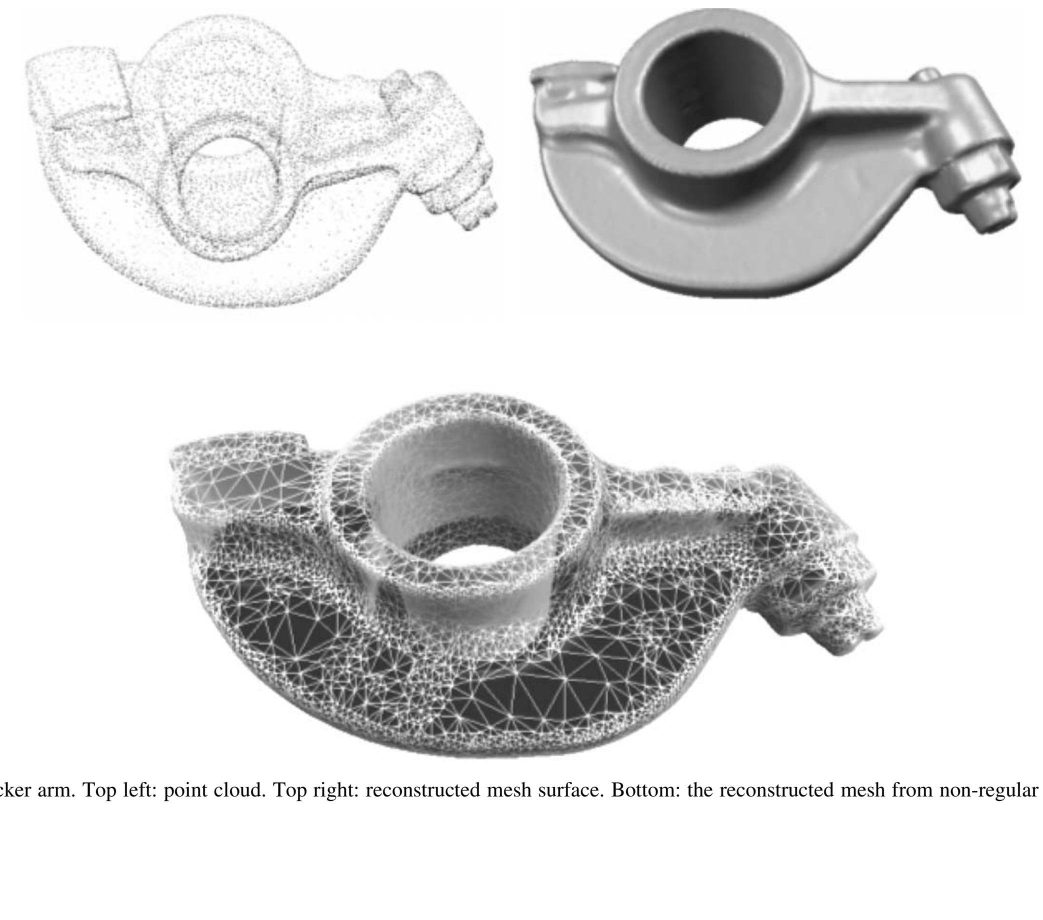
Fig. 10. Rocker arm. Top left: point cloud. Top right: reconstructed mesh surface. Bottom: the reconstructed mesh from non-regular point cloud.

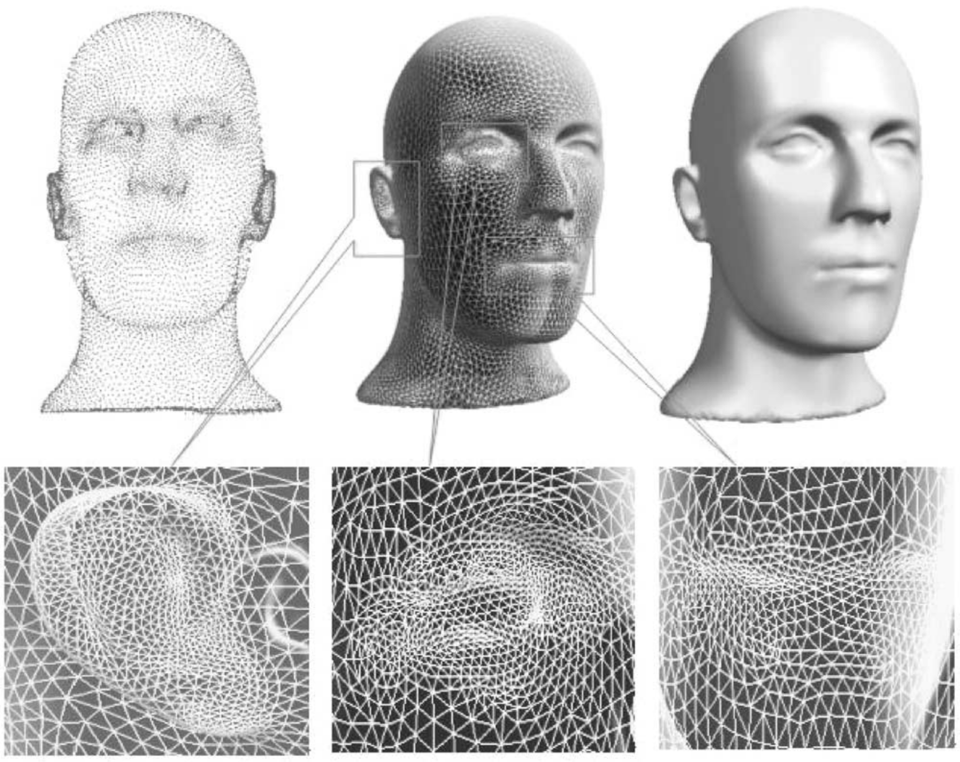
Fig. 11. Mannequin. Top left: point cloud. Top middle: reconstructed mesh. Top right: reconstructed mesh surface with light. Bottom left: zoom of ear. Bottom middle: zoom of eye. Bottom right: zoom of mouth.

Screwdriver, and Rocker arm are some CAD-like examples. The Rocker arm and mannequin head are examples of point clouds with non-uniform sampling density. The non-regular point cloud distribution of Rocker arm can be seen from the reconstructed mesh in Fig. 10. The point cloud for the mannequin head is highly non-uniform with much higher sampling density in the eyes, ears and mouth regions (see Fig. 11). Table 1 lists the experimental results of reconstructing these mesh surfaces using the *IPD* algorithm. It can be observed that the topological differences between the reconstructed surfaces and the sampled surfaces are small. The experiments were done on a PC with 500M Pentium and 128M memory.

Table 1
Evaluation of the *IPD* algorithm

Name	Genus	Times	#Vertices	#Triangles	Difference (e)
Dragon	0	152s	56194	112387	3
[a]Knee	2	85s	37888	75264	0
Cube_hole	5	21s	7672	15360	0
Fan_disk	0	56s	25893	51782	0
Screwdriver	0	67s	27152	54321	21
Rocker arm	1	25s	10044	20092	4
Mannequin	0	29s	11703	23403	1

[a] The difference formula (8) was used for the knee because it has boundaries. The knee possesses 2 pieces of boundaries and 512 boundary edges. The genus of the closed surface obtained by adding two new vertices to enclose the open ends is zero.

6. Conclusion

In this paper, we present a new region-growing mesh reconstruction algorithm, called the *IPD* algorithm. Most existing region-growing algorithms depend on user-specified parameter values to control the mesh reconstructing procedure; thus the quality and the topological error of the reconstructed meshes also depend on these parameter values. We formulate an intrinsic quantity for point clouds—the sampling uniformity degree at each sampling point—and devise a reconstruction algorithm driven by this measurement. Thus the algorithm is dependent solely on the point cloud itself and is independent of any user-specified parameters. Specifically, we define an influence region for each active edge, whose size is determined by the sampling uniformity degree at the two vertices of the active edge. A point is chosen from the active points within this influence region to form a new triangle with the active edge. We introduce a *weighted minimal length criterion* for selecting the new vertex. The criterion, which is based on harmonic energy, guarantees in theory that the constructed surface is closest to the surface of the sampled object. By utilizing an influence region that adapts to local sampling density and the weighted minimal length criterion for choosing new vertices, the IPD algorithm is able to reconstruct triangle meshes with small topological errors. The resulting meshes is therefore a good starting point to producing topologically correct reconstructed surfaces using some post-processing method, as in Ref. [11].

Acknowledgements

We thank Dr Ligang Liu of Microsoft Research Asia and Wei Chen of Zhejiang University for their great help. We are grateful to an anonymous referee for useful comments. The point clouds of the dragon, knee, screw driver and rocker arm are from Cyberware, and the fan-disk from University North Carolina. This work was supported by the National Natural Science Foundation of China (Grant No.60173034, No. 60021201 for Innovative Research Groups) and the Foundation of State Key Basic Research 973 Item (Grant No.2002 CB312101).

References

[1] Mencl E, Müller H. Interpolation and approximation of surfaces from three-dimensional scattered data points. State of the Art Reports, Eurographics'98, 98.; 1998. pp.51–67.

[2] Bernardini F, Bajaj C, Chen J, Schikore D. Automatic reconstruction of 3D CAD models from digital scans. Comput Geometry Appl 1999; 9(4–5):327–70.

[3] Boissonnat J-D. Geometric structures of three-dimensional shape reconstruction. ACM Trans. Graphics 1984;3(4):266–86.

[4] Edelsbrunner H, Mücke E. 3D alpha shapes. ACM Transactions on Graphics, Vol.(13), No.(1), 43–72

[5] Bajaj C, Bernardini F, Xu G. Automatic reconstruction of surfaces and scalar fields from 3D scans. Comput Graphics (Proc SIGGRAPH'95) 1995;August.

[6] Guo B, Menon J, Willette B. Surface reconstruction using alpha shapes. Comput Graphics Forum 1997;16(4):177–90.

[7] Veltkamp RC. Boundaries through scattered points of unknown density. Graphical Models Imag Process 1995;57(6):441–52.

[8] Kirkpatrick DG, Radke JD. A framework for computational morphology. Comput Geometry 1985;217–48.

[9] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. Comput Graphics(Proc SIGGRAPH'98) 1998;July:19–24.

[10] Amenta N, Bern M. Surface reconstruction by Voronoi filtering. Discrete Comput Geometry 1999;22(4):481–504.

[11] Adamy U, Giesen J, John M. Surface reconstruction using umbrella filters. Comput Geometry 2002;(21):63–86.

[12] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuelzle W. Surface reconstruction from unorganized points. Comput Graphics (Proc SIGGRAPH'92) 1992;71–8.

[13] Curless B, Levoy M. A volumetric method for building complex models from range images. Comput Graphics (Proc SIGGRAPH) 1996;303–12.

[14] Bernardini F, Bajaj C, Chen J, Schikore D. Triangulation-based object reconstruction methods. Proceedings of SCG'97 (ACM Symposium on Computational Geometry); 1997. 481–484.

[15] Boissonnat J-D, Cazals F. Smooth surface reconstruction via natural neighbor interpolation of distance functions. Proceedings of SCG'00 (ACM Symposium on Computational Geometry); 2000. 223–232.

[16] Bernardini F, Mittleman J, Rushmeier H, Silva C, Taubin G. The ball-pivoting algorithm for surface reconstruction. IEEE Trans Visualization Comput Graphics 1999;5(4):349–59.

[17] Huang J, Menq CH. Combinatorial manifold mesh reconstruction and optimization from unorganized points with arbitrary topology. Comput-Aided Des 2002;(34):149–65.

[18] Petitjean S, Boyer E. Regular and non-regular point sets: properties and reconstruction. Comput Geometry 2001;(19):101–26.

[19] Eck M, DeRose T, Duchamp T. Multiresolution analysis of arbitrary meshes. Proc Siggraph'95 1995;173–80.

[20] O'Rouke J. Triangulation of minimal area as 3-D object models. Proc Int Joint Conf Artif Intell 1981;81:664–6.

[21] Epprtein D. Approximating the minimum weight triangulation. Discrete Comput Geometry 1994;(11):163–91.

[22] Dickerson MT, McElfresh SA, Montague MH. New algorithms and empirical findings on minimum weight triangulation heuristic. Proceedings of the 11th Annuual ACM Symposium on Computational Geometry; 1995. 238–247.

Hong-Wei Lin is a PhD student at the Institute of Computer Images and Graphics in Zhejiang University, China. He received his BS degree from Department of Applied Mathematics at Zhejiang University in 1996. He worked as a communication engineer from 1996 to 1999. His current research interests are in computer aided geometric design, computer graphics, and image process.

Chiew-Lan Tai received the BSc degree in mathematics from the University of Malaya, the MSc degree in computer and information sciences from the National University of Singapore, and the DSc degree in information science from the University of Tokyo. She has been an assistant professor in the Department of Computer Science at Hong Kong University of Science and Technology since 1997. Her research interests include geometric modelling and computer graphics.

Guo-Jin Wang is a professor and superviser of doctoral students in the Department of Mathematics and the Institute of Computer Images and Graphics, Zhejiang University, Hangzhou, China. He received his BSc and MSc in Mathematics from Zhejiang University. His teaching and research activities are concerned with applied mathematics, computer aided geometric design, computer graphics and geometric modeling. Between 1991 and 1993, he was a visiting researcher at Brigham Young University, USA. He also was a visiting researcher at Hong Kong University of Science and Technology, in 2001 and 2003. He has published numerous papers on CAGD and CG, and is the co-author of the book, CAGD, published by China Higher Education Press, Beijing and Springer-Verlag, Berlin Heidelberg.