International Journal of CAD/CAM Vol. 7, No. 1, pp. 103~112 (2007)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

A Two-Phase Approach of Progressive Mesh Reconstruction from Unorganized Point Clouds

Hongxin Zhang, Hua Liu, Wei Hua* and Hujun Bao
State key lab of CAD&CG, Zhejiang University, Hangzhou, China {zhx, sun_day, huawei, bao}@cad.zju.edu.cn

Abstract – This paper presents a practical approach for surface reconstruction from unoriented point clouds. Instead of estimating local surface orientation, we first generate a set of depth images from the input point clouds, and a coarse mesh is then generated based on them by space carving techniques. The resultant mesh is progressively refined by local mesh refinement and optimization according to surface distance measure. A manifold mesh approximating the input points within an given tolerance is finally obtained. Our approach is easy to implement, but has the ability to outputs high quality meshes in different resolutions. We show that the proposed approach is not sensitive to several types of data disfigurement and is able to reconstruct models robustly from variance input data.

***Keywords*: Surface reconstruction, Progressive mesh, Differential mesh deformation**

1. Introduction

In this paper, we present a 2-phase solution for the surface reconstruction problem: how to reconstruct a high quality mesh surface from 3D point clouds without knowing normal information. As the wide range of modeling applications rely on surface reconstruction from scatted data points, many automatic reconstruction algorithms were developed in the past two decades. Most of them, however, require sufficient sampling, and even some of them require additional accurate surface normal estimations. Unfortunately, the real input data does not always satisfy these precision requirements. Moreover, large amount of imperfect factors, e.g. fake points, incomplete data and different types of noise (see Fig. 1), may interfere with the reconstruction process. As pointed out by [18], it is still a difficult problem to reconstruct manifold surfaces from unoriented point sets.

Previous dominant approaches mainly apply the theories of Voronoi diagrams from computational geometry or use of volumetric reconstruction techniques. Most Voronoi based approaches reconstruct mesh by using the input points as positions of vertices directly. Therefore it is difficult to produce a smooth and manifold surface from a noisy and poorly sampled point cloud. While volumetric methods, on the other hand, tend to define distance functions to the point samples and then retrieve level-set surfaces. They suffer the problems of high memory storage consumption for extracting shape details and unstable normal estimations when densely sampling around sharp or thin features.

We propose a novel two-step reconstruction approach in this paper. Different from previous approaches, the main advantage of our approach is that no normal information is required. We leverage visibility information inferring surface orientation instead of directly estimating surface normal vectors for the purpose of robust surface reconstruction. In a human vision system (HVS), shapes are mainly observed and recognized by silhouette and depth. It is known that a visual hull can be well constructed from silhouettes, and therefore global shape orientation can be roughly learnt from it. Meanwhile depth quantities provide accurate information, which can be used to deal with concave aspects. So the major motivation of our work is to analog the shape reconstruction process of HVS for reconstructing topologically correct shape by taking a set of depth snapshots (see Section 4).

Another concerning of our work is geometry details of the target models. An ideal reconstruction process should efficiently reconstruct shape details as much as possible from noisy input. Volumetric approaches mainly use hierarchical space subdividing to increase reconstruction resolutions for extracting these details. However, it is a space consuming solution. Contrary, our contribution is to directly refine mesh surfaces so as to capture high frequency details via surface up-sampling. Therefore our surface based refinement approach consumes smaller storage spaces and can reach higher resolution than volumetric based ones.

2. Related work

We briefly review typical approaches on surface reconstruction from unoriented points and related techniques in this section.

Voronoi diagrams based methods, e.g. the "crust" algorithm [2,3] and [6,12,13], reconstruct surface boundary by erasing those cells that do not belong to the volume bounded by the sampled surface.

These methods are good at processing surfaces with high

*Corresponding author:
Tel: +86-(0)571-88206679
Fax: 186-(0)571-88206680
E-mail: huawei@cad.zju.edu.cn

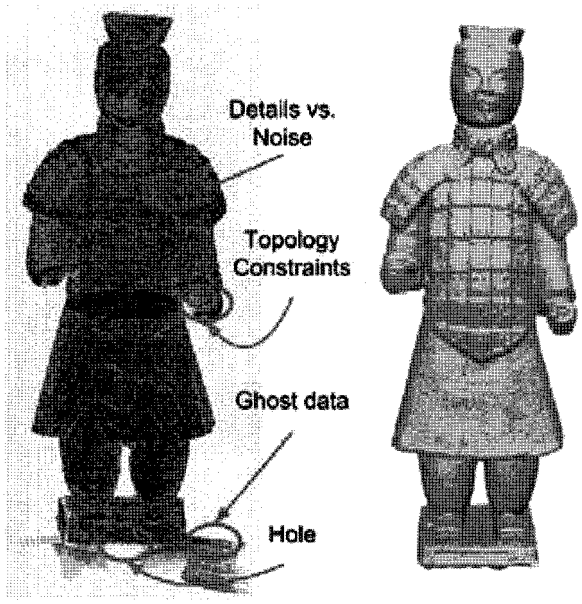


Fig. 1. Terra-cotta warrior. The left figure illustrates the challenging problems of real scanned data. And the right model is a reconstructed result by our proposed method.

genus. And they work fairly well when the sampling data are sufficiently dense and smooth. Many other methods [5,27] tend to generate surfaces in an incremental manner. In all those methods, however, the local topology may not be correct and holes may appear due to under-sampling. Recent work in this direction such as [23] can now dealing with noisy input data.

Large amount of studies cast the surface reconstruction problem as a data fitting mission for tackling noise. A specific distance-like measure in general is formulated to find a best fitting one. These methods in this field can be roughly classified into parametric-based and implicit surfaces-based ones.

Parametric-based approaches [16,17,28] assume that the underlying topology of target surface is known, which is homeomorphism to a given parametric domain. And they mainly apply compact surface descriptions. Specifically, several recent-developed methods [29] deform an initial surface along an energy field induced by the input points. Such kind of approaches is good for produce smoothed surfaces. However, there exist two drawbacks when applying them: the potential difficulties for generating initial topological-correct surface in high genus cases, and finding appropriate surface fitting parameters.

Approaches such as [10,11,14,19,25] reconstruct surface based on implicit functions which indicate the outlier and inside of the target surface. These algorithms depend on relatively accurate estimations of surface orientation and uniformed sampling of input data, which actually are crucial for the real scanned data. In addition, they may introduce topological artifacts for the data containing poorly aligned scan patches due to the distance-like function definition.

In vision research, shape from X, e.g. from shading, silhouette and stereopsis, are relevant topics of surface reconstruction. Visual hulls [21] and photo hulls [20] are typical approaches of volumetric carving to extract shape geometry from image information. Although the first phase of our approaches is also a space carving one. Different from them, our input are point cloud data which maintain more accurate shape information, and depth images are captured by a virtual camera for providing plausible surface orientations.

The latest trend for combining vision techniques with surface reconstruction is the graph cut optimization, e.g. [18, 26]. It is a global optimization technique to efficiently solve image and discrete volumetric segmentation problems by re-formulating them as a minimal cut problem of a spatial graph structure [9]. Unfortunately, implementing these methods usually require discrete volumetric descriptions. Therefore it is difficult to reach high reconstruction resolution because of cubic increased space consumption.

Surface tessellation by using Marching Cubes [22] and Marching Tetrahedra [7] is a critical step in surface extraction. Besides geometric accuracy may drop after this process, result meshes are normally in low quality. These meshes contain lots of thin and elongated triangles with ugly topological connectivity. It is inconvenient for succeeded modeling applications. Thus remeshing techniques always act as a post-processing step for remedying this disadvantage. Recent advances of remeshing [1] are mainly focused on creating desired nice meshes after a high resolution surface has been already known. Our methodology principle, alternatively, is to directly reconstruct a satisfied mesh both in high mesh quality and geometric accuracy.

3. Method Overview

Our algorithm consists of following two phases, i.e., coarse model generation and progressive mesh refinement. An example of the whole reconstructing process is illustrated in Fig. 2.

Phase I is *initial mesh generation*. In this phase, we generate a coarse model, which is homeomorphism to the underlying target surface, as the input data for the following process. As illustrated in Fig. 2, the input points are observed by a virtual camera from different positions and directions, resulting a set of depth images. Then a confidence map is calculated according to these depth images. Necessary user interactions are also involved to guarantee the correctness of the output surface. When the space confidence map is built up, we can easily extract the initial mesh surface. Details are presented in Section 4.

Phase II is *progressive mesh refinement*. In this phase, an error function is introduced to measure the distance between the reconstructed surface and input points. Then triangles in relatively large error are split. And new generated vertices are moved to the appropriate position according to input points so as to decreasing distance error. A remeshing process is also required in this phase. In Section 5, we will describe the refinement process in detail.

4. Coarse Mesh Generation

We present the process of generating coarse model in this

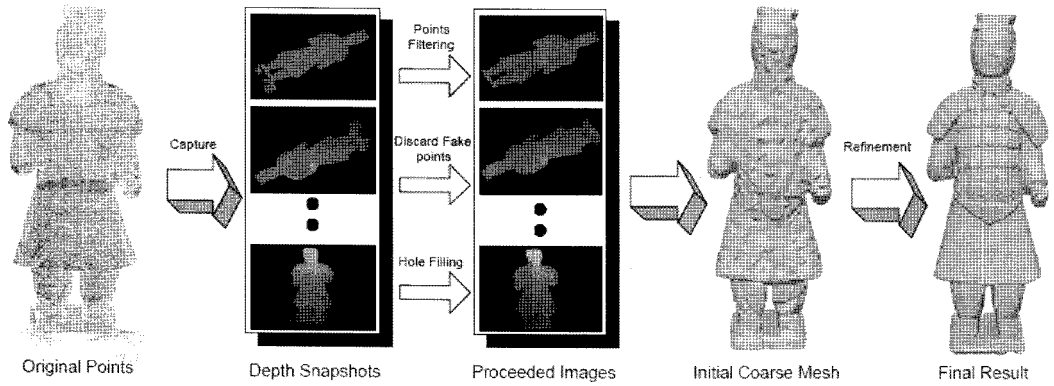


Fig. 2. Illustration of method overview.

section. The whole process consists of taking depth snapshots, data cleaning, confidence map computing and triangular mesh extraction, consequently.

4.1. Depth snapshots

In our approach, the input data is a set of 3D points $\boldsymbol{P}$ without knowing orientation information. Initially, all input points $P=\{p_i \mid p_i \in \Re^3, i=1,2,\ldots N\}$ are normalized by scaling so that they fall into the cube $[-1,1]^3$.

To capture geometric information, we take a group of depth images for $\boldsymbol{P}$ by a virtual camera from different view points and directions which are presented as $\{(\boldsymbol{vp}_i, \boldsymbol{vd}_i) \mid i = 1,2,\ldots m\}$. In our implementation, data $\boldsymbol{P}$ is rendered in a 3D environment (based on OpenGL API) by using *orthomorphic projection* with depth test. For each point $\boldsymbol{p}_i$, its color is evaluated according to its depth value in the camera coordinate system. That is $Color(\boldsymbol{p}_i) = (z(\boldsymbol{p}_i), z(\boldsymbol{p}_i), z(\boldsymbol{p}_i))$ with

$$z(p)=1.0-\frac{Depth(p)-NearestDepth}{FarthestDepth-NearestDepth}.$$

Here the function $Depth(\boldsymbol{p})$ returns the z value of point $\boldsymbol{p}$ in the camera coordinate system. Larger $z(\boldsymbol{p})$ means nearer to the nearest plan. And we set background color of these images as black, i.e., $z(\boldsymbol{p}) = 0$. A set of snapshots which indicates depth information are then obtained. We call them *depth snapshots*.

The sampling manner $\{(\boldsymbol{vp}_i, \boldsymbol{vd}_i) \mid i = 1, 2,\ldots,m\}$ of above virtual acquiring procedure is quite flexible. The only constraint of a sampling manner is the whole visible hull of a give data shall be covered. Commonly, we set sparse view points on the bounding sphere of data $\boldsymbol{P}$. The view points in our prototype system are selected at $\boldsymbol{vp}_i = (\cos(i\theta), \sin(i\theta), 0)$ with $\theta = 2\pi/(m\text{-}2)$. And two additional viewpoints are top and bottom the sphere. The orientations of camera is then set to be aimed at the sphere center, i.e., $\boldsymbol{vd}_i = -\boldsymbol{vp}_i$. Hence the sampling process can be executed automatically. These depth snapshots can also be specified by user, to reduce the amount of images. In addition, few more depth snapshots may be manually captured to emphasize sampling on specific parts.

Overall, capturing depth snapshots is very efficient. In all our experiments, 8~15 depth snapshots are sufficient for generating the coarse mesh, since depth information is more accurate than solely visibility mask. And the average time to process one depth snapshot is less than one second.

The advantage of using depth snapshots is twofold. One is that visibility and orientations of underlying target surface are well captured by these depth images. The other is that concave shape features can be more accurately described and less images are required comparing with the silhouette or contour based image descriptions for space carving, e.g. the visual hulls.

4.2. Confidence Map Estimation

Once a set of snapshots in hand, we compute a *confidence map* in vicinity of these processed point samples, similar to many volumetric based approaches (e.g. [11]). We compute these confidence values as a distance-like function $\varphi: v \to c \in [-1,1]$ over the voxels $v \in V$ in a volumetric grid, where c can be viewed as the pseudo-distance of a voxel to the visible boundary of underlying solid respected to the point cloud $\boldsymbol{P}$. The map φ represents confidence values which is inside (negative) or outside (positive) of the unknown watertight surface.

Its calculating is performed in a cumulative way. Simultaneous to the sampling of depth snapshots, the value of confidence map is updated when a new depth image is added. For the k-th snapshots I_k, let $d_k(v)$ be the value to indicate whether the voxel v is the distance between v and the ios-surface (see Fig. 3). It is calculated by the following formula:

$$d_k(v) = z_k(v) - c_k(v).$$

In Eqn. (2), the function $c_k(v)$ returns the first channel value of the corresponding 2D pixel color of point v projected in the k-th image. The value z_k stands for the depth of v computed by Eq. 1. Let m be the number of the sampling images, the confidence map $\ddot{o}(v)$ is calculated as follow:

$$\varphi(v) = max\{d_k(v) k = 1,2,\ldots,m\}.$$

4.3. Data Cleaning

To obtain a well-defined confidence map, data cleaning of point clouds is an essential pre-process step before that.

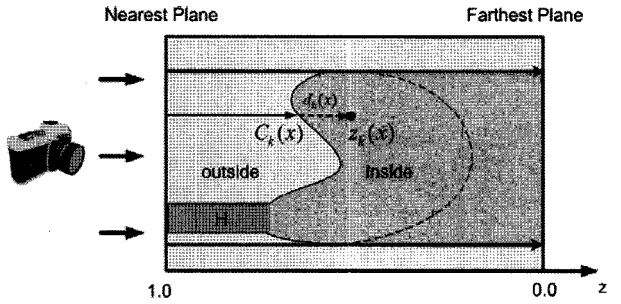


Fig. 3. Depth snapshots and confidence map. The dark gray region H is a hole covered by the white color.

When using optical scanning device, scanned data may contain ghost aspects and incomplete parts as shown in Fig. 4. They cause difficulties and unsatisfied results. Fortunately, a set of 2D depth image has been already captured. We apply following image based operations to avoid above addressed problems.

Fake Points. Note that these fake points, which should be discarded, can be well observed from depth snapshots. We therefore paint all the pixels which correspond to fake data in black. And simultaneously, corresponding points are removed from the point set P. After that, we regenerate all depth snapshots for the lack of sampling objects in reasonable visible size. Here, notice that all the painted fake points in the snapshots should be clearly observed and do not occlude any actually data (see Fig 4).

Incomplete parts or holes. As the input points are distributed sparsely in 3D space, pixels in depth snapshots may be discontinuous, and may not group into regions (see Fig. 4). Therefore, when processing snapshots I_i, we use an additional rendering pass in the same view position to generate a smoothed snapshots I_i^l by increasing the point size, so as to make continuous regions. We can cover all the small under sampling parts simply by this method.

Regarding for *large incomplete aspects or holes*, following operations are performed. As shown in Fig. 5, the foot part of the baby model contains hole, where the sampling points are unavailable. If it is not well fixed, the confidence map in the next section will not be correctly calculated. Thus we simply fill this hole in white color. To achieve this goal, the region represent the hole is painted in white color, i.e. by setting the confidence value to be 1. That means the distance function $d_k(x)$ of the points projected into the region of hole are not contributed to the $D(x)$ in the k-th image. Although it is a conservative estimation by doing so. The final confidence values of these points, however, are corrected by other snapshots from different view directions.

Topological correctness control. As mentioned above, we covered all the small gaps in snapshots by increasing the rendering point size directly. But sometimes, this straightforward method may lead to merging of unexpected disconnected regions. For this reason, we generate a binary image for each depth snapshots by setting value 1 for these pixels with non black color and 0 for black pixels. Then several steps of a morphological dilation operator are performed to obtain a visibility mask . In this step, some special tagged pixels may be inserted into these binary images to constrain the dilation regions (see Fig 4, the red pixels are tagged pixels).

4.4. Mesh extraction

The initial coarse mesh is extracted as the zero level-set

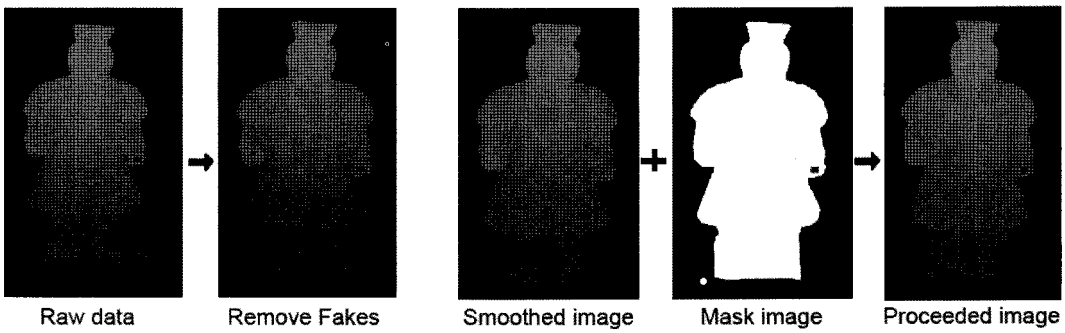


Fig. 4. Data cleaning for the terra-cotta warrior data.

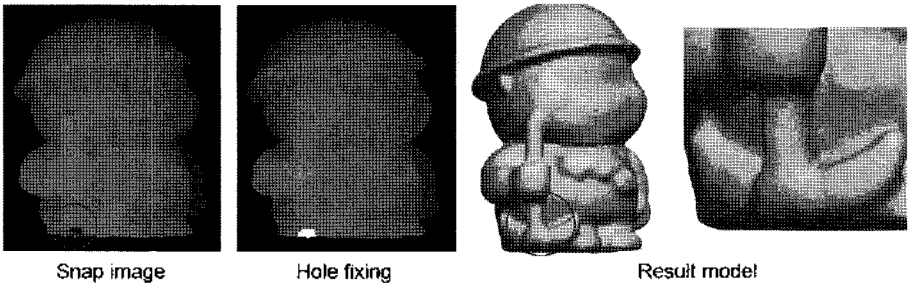


Fig. 5. Fixing a hole in the rag baby model. The observed hole is painted with white color.

of the confidence map. To generate this mesh, we employ a marching cubes variant with a lookup table that resolves ambiguous cases [24]. As we only need a mesh in low resolution, the sampling voxel size can be relatively large. This makes satisfied time and space efficiency at this stage.

5. Progressive Mesh Refinement

Once an initial mesh is obtained, the left issue is to extracting geometry details from the original point cloud. Inspired by previous surface based approaches, our solution in principle is a data fitting one. That is to achieve low approximating error as well as good shape quality. Precisely, this optimization principle can be written as:

$$\min_{S}[E_m(S;\boldsymbol{P})+wE_s(S)], \tag{4}$$

where E_m is a distance measure to control reconstruction precision, and E_s is a smoothness term with weight w to prevent self-intersection problem during fitting. However, the optimization formulation is always non-linear and is seldom in close form, i.e., the problem cannot be solved directly. Further more producing high quality meshes are additional requirement for solid modeling applications.

Based on above concerning, we propose an iterative algorithm for detail extracting. The detailed pseudo code is listed in Fig. 6. Decreasing E_m is achieved in a greedy optimization way. In each approximation iteration step, candidate vertices are located at the positions where large local approximation error are occurs. And then these vertices are inserted into the target mesh to increase approximation accuracy. A successive remeshing procedure is performed to provide good mesh quality and to control sampling resolution.

Algorithm 5.1: ProgMeshRefine($\mathcal{S}_0, \mathcal{P}, \epsilon$)

```
procedure InsertCandidateVert(S, P)
 for each p_i ∈ P
   do { f_j ← FindNearestFace(S, p_i);
        CS(f_j) ← CS(f_j) ∪ {p_i};
 for each f_j ∈ S
   do { p_i ← argmax_{p∈CS(f_j)} d_{p,f_j};
        ε_j ← d_{p_i,f_j};
 ε̄ ← AverageError();
 for each f_j ∈ S
   do if ε_j > ε̄ InsertVertex(f_j);
 return (S);

main
 S ← S_0; k ← 0;
 while E_m(S, P) > ε
   do { InsertCandidateVert(S, P);
        Remeshing(S, l_k);
        k ← k + 1;
 return (S);
```

Fig. 6. The pseudo code of progressive mesh refinement.

5.1. Insert candidate vertices

In this paper, we utilize the Hausdorff distance function to measure the errors between approximating mesh surface S and the cleaned point cloud $\boldsymbol{P}'$. That is

$$E_m(S,P')=\max_{p_i\in P'}\{\min d_{p_i,f}\} \tag{5}$$

where $d_{pi,f}$ denotes the Euclidean distance between the point p_i and a triangle f of surface S.

According to the distance definition, we use following procedure to find the candidate vertices to improve fitting accuracy (please see Fig. 6). Firstly, for each point $p_i \in P'$, find the nearest face $f_j \in S$, and calculate the distance value $d_{pi,fj}$. At the same time, the point p_i is assigned as one of the corresponding point of f_j. Secondly, for each face $f_j \in S$, find the point p_j with max distance value in all the corresponding points of f_j . Then the distance value is saved as the local error e_j of face f_j and the point p_j is stored as the target point of f_j. If there is no corresponding point for f_j, the local error ε_j is set to zero. Calculate the average local error $\bar{\varepsilon}$. Finally, insert a new vertex into the face f_j if $\varepsilon_j > \bar{\varepsilon}$ and move the new generated vertex to the specific position.

As illustrated in Fig. 7, three situations will occur when inserting new vertices. We call them: *face subdivision*, *edge cutting* and *vertex moving*, respectively. The choice of splitting operation depends on the distance between f_j and its target point p_j. If the projection of p_j onto the face f_j is inside the triangle fi, then face subdivision is applied. Otherwise, calculated the distance $d_{p,ek}$ between p_j to the edges e_k (k = a, b, c) of f_j and the distance $d_{p,vl}$ between p_j to the vertices v_l(l = a, b, c) of f_j. If $d_{p,ek} < d_{p,vl}$, the edge cutting is applied, otherwise we only move the vertex of f_j, which is nearest to the point p_j.

The position of the new generated vertex v_i is calculated as follows:

$$v_i=v_i+t(p_i-v_i)+(1.0-t)\frac{1}{k}\sum_{j=1}^{k}(p_j-v_j), \tag{6}$$

where t is a tension parameter and k is the number of adjacent vertices of v_i. The new position of v_i is determined by two factors: (i) the position of its corresponding point p_i; and (ii) the affection from the adjacent vertices.

5.2. Remeshing

The above naive distance-based updating strategy can only generate resultant meshes in low quality. The refinement results are a little bit bumpy, and long thin triangles are appeared because of the face spilt operations. Hence an additional remeshing step is carried out after candidate vertices are inserted in each updating cycle. The remeshing approach we adopted is a variant implementation of [27]. Briefly speaking, this remeshing technique is an iterative procedure. In each processing pass, given a target length l, a remeshing step consists of four consequent operations, namely OperatorS, OperatorC, OperatorV and OperatorT respectively.

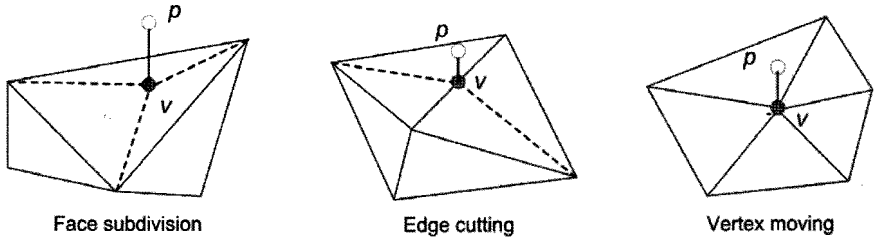


Fig. 7. Insert candidate vertices.

- **OperatorS** : For each edge e, if *length*(e) > 4/3 l, insert the midpoint of e.
- **OperatorC** : For each edge e, if *length*(e) < 4/5 l, collapse e to its midpoint.
- **OperatorV** : For each edge e, flip it if the average valence of its four adjacent vertices (in its adjacent triangles) can approach nearer to 6.
- **OperatorT** : For each vertex v, re-compute its position on the surface by tangential smoothing.

In our approach, we use dynamic target edge length,

$$l_k = \lambda_k l_{k-1}, k = 1,2,\ldots \tag{7}$$

with λ_k to be a tuning factor at k-th refinement pass for mesh reconstruction. Let l_0 be the average edge length of the initial coarse mesh S_0. Note that S_0 is generated by the marching cube. There are lots of long thin and zigzag triangles on S_0. Therefore l_1 is chosen to be slightly longer than l_0 to achieve better mesh quality and smoothing effects simultaneously. In our experience, $\lambda_1 = 1.2$ is appropriate. And in the successive several passes, $_k$ is set to be 1.0 until the error value is stable. The major motivation is based on the observation that the distance between refined mesh M_k and target data points is quit large in the first several passes. Then we turn to choose $\lambda_1 \in [0.7, 1.0)$ to achieve higher mesh resolutions.

One crucial issue in the remeshing algorithm is the topology preserving. When "OperatorC" is applied in refinement procedure, it may potentially change the genus of shape and produce degenerate connectivity. Therefore, an addition function for connectivity checking is applied before performing the "OperatorC" in our implementation.

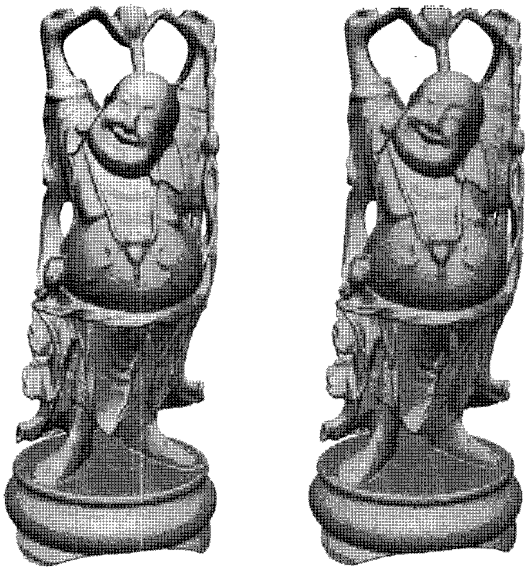
The coupling of mesh refinement and remeshing implies a mesh surface fitting procedure indeed. The mesh refinement part takes care of position constraints. While the remeshing step provides the smoothing function to prevent ugly approximated meshes. In addition, the sampling resolutions of meshes are well controlled by remeshing.

Although the remeshing step may slightly increase the approximate error mainly due to the "OperatorT". The trend of the whole process will still decrease error E gradually. In our approach, a new surface is generated after each refinement pass. At last, besides the final reconstructed surface, we can obtain a series of well constructed meshes in different resolutions.

6. Implementation and Results

We have performed our method on several challenging point cloud data (e.g. Fig. 8, 9, 10 and 11). We also scanned two models, the warrior (Fig.1) and the rag baby model (Fig.12) for testing the robustness of our proposed algorithm by using a FastSCAN hand-held laser scanner.

In all our experiments, the resolution of depth snapshots



(a) Global remeshing (b) Local remeshing

Fig. 8. Happy Buddha.

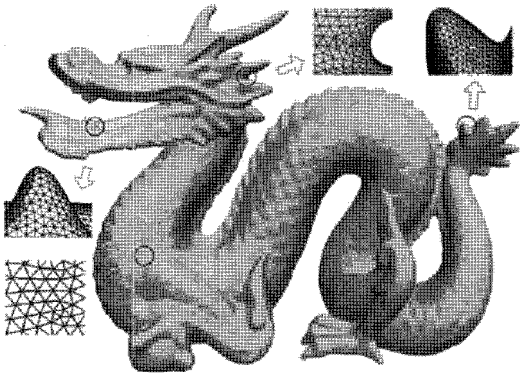


Fig. 9. Dragon.

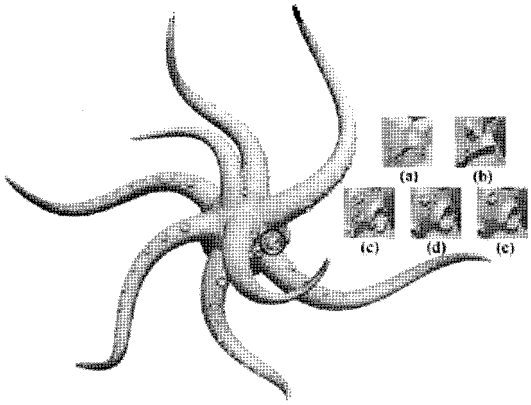


Fig. 10. Octopus.

is 800 × 600 to capture thin or flam features in point clouds. The confidence map is defined in the cube $[-1,1]^3$ and the sampling step is set to be 0.02 unit along each axis.

The statistic data are listed in Table 1. All the experimental data were collected on a PC equipped with Dual Intel Xeon 2.4 GHz CPU processor and 3 GB memory. In Table 1, it can be observed that the running time of Phase I mainly depend on the number of depth snapshots due to the unified rasterization resolution. This feature enables us to process huge data efficiently.

In Phase II (cf. Table 1), our approach normally consumes much longer time for later several passes, since the resolutions of approximating meshes are increase. To overcome this problem, a local updating trick can be used in our remeshing procedure. That is we only perform remeshing around nearby regions where triangles are recently refined. By using this implementation trick, the whole processing time in our experience is nearly half of the global remeshing version. But results will be slightly worse than global updating version both in mesh smoothness and topological connectivity. A result comparison is demonstrated in Fig. 8.

Our presented results exhibit the abilities of our robust reconstruction method. In Figs. 1, 8, 9 and 10, models are in high shape complexity and full of details. These models can be reconstructed by using moderately number of depth snapshots.

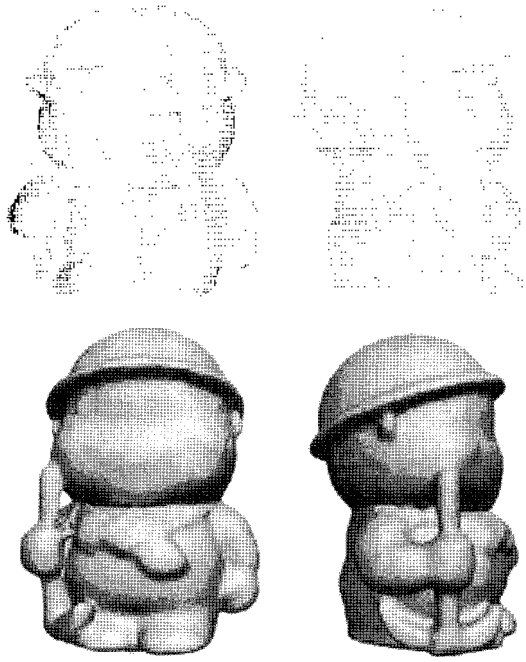
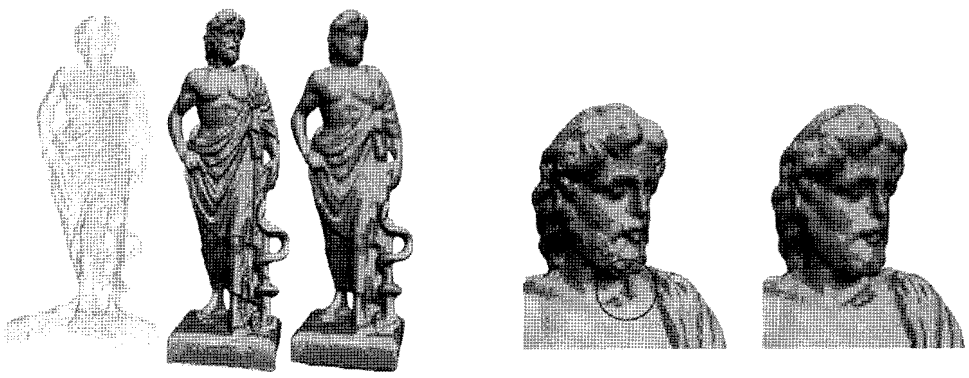


Fig. 12. The rag baby model. The upper row shows the scanned point clouds viewed from front and left-side, respectively. The lower row shows the reconstructed surface.

Meanwhile the final accurate outputs are in high quality in terms of evenly sampling and near regular connectivity, as illustrated in Fig. 9. Therefore the outputs of our approach can be directly utilized in various computer graphics applications, such as differential mesh modeling [30] and point-based graphics.

In all our examples, data details are precisely recovered in an evolution manner. For example, in Fig. 10, acetabula of the octopus are appeared gradually when reconstruction resolution is increased. Reconstructing leg ends of the octopus and horns of the dragon are not easy for some other methods.

Comparing with stat-of-the-art reconstruction algorithms, our algorithm still shows outstanding performance. Fig. 11(a) demonstrates a comparison with the Poisson surface reconstruction (PSR) algorithm [19], one of the best volumetric approaches



(a) (b)

Fig. 11. Algorithm comparisons of the Greek data. (a) The left image is the original scanned data. The middle image is our reconstructed result, and the right one is obtained by Poisson surface reconstruction. (b) The left model is reconstructed by the tight cocone algorithm. And the right one is our reconstruct result.

Table. 1. Performance statistics

Input Model			Phase I		Phase II		Output #triangles	Em(P', S')
Name	Figure	#points	#snapshots	run(sec)	#passes	run(sec)		
Octopus	Fig. 10	149,666	13	11.5	15	425.0	503,084	5.5e-5
Warrior	Fig. 1	918,257	8	8.9	16	640.2	908,118	5.9e-4
Dragon	Fig. 9	437,645	9	9.3	16	632.3	875,968	7.2e-4
Buddha	Fig. 8	543,652	10	9.8	14	1008.2	1,225,162	1.8e-4
Greek	Fig. 11	102,352	15	17.0	17	290.8	182,300	3.6e-5
Rag baby	Fig. 12	215,336	12	10.8	15	340.0	206,614	5.8e-4

developed recently. In this example, we use default parameters for PSR and no additional normal information are provided. It is obvious that our reconstructed result (the middle image of Fig. 11(a)) contains more details than [19] (the left image of Fig. 11(a)). The major reasons are PSR requires initial estimation of surface orientation and it contains a PDE based optimization procedure to diffuse normal deviations. We also compared our algorithm with the tight cocone (TC) [12], a typical Voronoi-based approach. As illustrated in Fig. 11(b), the result generated by TC (the left image) still contains topological failures (below the chin) even the input data are quite clean. While our algorithm produces correct result (the right image of Fig. 11(b)) due to the visibility-based reconstruction strategy.

The terra-cotta warrior (Fig. 1) and the rag baby model (Fig. 12) demonstrate the ability of data repairing. To guarantee the result surface to be manifold, we fix the holes and discard all fake points during the depth sampling step. In Fig 1, the base plat and several insufficient sampled part of the warrior are well repaired, and unnecessary parts are discarded. Regarding for the rag baby model illustrated in Fig. 13, the inner side of the stick is unreachable by our hand-held laser scanner.

7. Discussion and Future Work

In this paper, a hybrid approach is presented for reconstructing surfaces from unorganized points without knowing surface orientation information. Actually, two sampling strategies are seamlessly coupled in our approach. In the first phase, uniform sampling based on visibility ensures correct surface topology and implicitly provides robust surface orientation estimation. User interactions are also able to be conveniently performed for data repairing in this stage. In the second phase, an area-equalizing surface sampling is carried out, which is guided by surface distance measure. Therefore high quality mesh surfaces with elaborate details are well recovered.

Our method is limited in recovering visible parts of a closed object, which is identical to the human's perception of what a surface is in most cases. As mentioned in Section 4.1, our sampling principle is to cover all visible parts of input data. However, this rule may not work well if we merely move and rotate a virtual camera on the bounding sphere of a complex model with invisible parts, e.g., a seashell. In this situation, a possible solution is to divide a given point cloud into several parts, then to treat them separately, and finally combing separated parts.

Several adaptive strategies may be useful to enhance the ability of our reconstruction approach. The adaptively sampled distance fields [15] can be used instead of the uniformly sampling in Phase I. And also, an adaptively remeshing method may be adopted according to specific properties of the input point clouds, such as sharp features and/or curvature measure. This improvement can reduce the number of triangles. To achieve even higher reconstruction quality, we tested a MLS projection technique for extracting details, which is similar to the final projection step addressed in [29]. It is promising direction to combine local surface fitting for the purpose of denoising. In addition, the Edge-sharpener technique [4] can be integrated into the mesh refinement procedure to enhance features.

Acknowledgements

Models are courtesy of Cyberware, Stanford University, Max-Planck-Institut fur Informatik and AIM@SHAPE shape repository. This project is supported in partial by the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (No. 705027), 973 Program of China (No.2002CB312102), and NSFC (No.60505001).

References

[1] Allicz, P., Ucelli, G., Gotsman C. and Attene M. (2005), Recent advances in remeshing of surfaces. Technical report, Part of the state-of-the-art report of the AIM@ SHAPE EU network.

[2] Amenta, N., Bern, M. and Kamvysselis M. (1998), A new voronoi-based surface reconstruction algorithm, ACM SIGGRAPH '95 Conference Proceedings, 415-421.

[3] Amenta, N., Choi, S. and Kolluri, R. K. (2001), The power crust, Proceedings of the sixth ACM symposium on Solid modeling and applications, 249-266, ACM Press.

[4] Attene, M., Falcidieno, B., Rossignac, J. and Spagnuolo, M. (2003), Edge-sharpener: A geometric filter for recovering sharp features in uniform triangulations, In Eurographics Symposium on Geometry Processing, Aachen, Germany, June 2003.

[5] Bajaj, C. L., Bernardini, F. and Xu, G. (1995), Automatic reconstruction of surfaces and scalar fields from 3d scans, ACM SIGGRAPH '95 Conference Proceedings, 109-118, ACM Press.

[6] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C. and Taubin G. (1999), The ball-pivoting algorithm for surface

reconstruction, *IEEE Transactions on Visualization and Computer Graphics* **5**(4), 349-359.

[7] Bloomenthal, J. (1994), An implicit surface polygonizer, In P. Heckbert, editor, *Graphics Gems IV*, 324-349, Academic Press, Cambridge.

[8] Botsch, M. and Kobbelt, L. (2004), A remeshing approach to multi-resolution modeling, Eurographics/ACM SIGGRAPH SGP '04 Proceedings, 185-192, ACM Press.

[9] Boykov, Y. and Kolmogorov, V. (2003), Computing geodesics and minimal surfaces via graph cuts, Proceedings of IEEE ICCV 2003, 26–33.

[10] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C. and Evans, T. R. (2001), Reconstruction and representation of 3d objects with radial basis functions, ACM SIGGRAPH '01 Conference Proceedings, 67-76, ACM Press.

[11] Curless, B. and Levoy M. (1996), A volumetric method for building complex models from range images, ACM SIGGRAPH '96 Conference Proceedings, 303-312.

[12] Dey, T. K. and Goswami, S. (2003), Tight cocone: a water-tight surface reconstructor, Proceedings of the eighth ACM symposium on Solid modeling and applications, 127-134, ACM Press.

[13] Edelsbrunner H. and Meucke E. P. (1994), Three-dimensional alpha shapes, *ACM Trans. Graph.* **13**(1), 43-72, 1994.

[14] Fleishman, S., Cohen-Or, D. and Silva C. T. (2005), Robust moving least-squares fitting with sharp features, *ACM Trans. Graph.* **24**(3), 544-552.

[15] Frisken, S. F., Perry, R. N., Rockwood, A. P. and Jones T. R. (2000), Adaptively sampled distance fields: a general representation of shape for computer graphics, ACM SIGGRAPH '00 Conference Proceedings, 249-254, 2000.

[16] Hoppe, H., DeRose, T., Duchamp, T., McDonald J. and Stuetzle W. (1992), Surface reconstruction from unorganized points, ACM SIGGRAPH '92 Conference Proceedings, 71-78.

[17] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle W. (1993), Mesh optimization, ACM SIGGRAPH '92 Conference Proceedings, 19-26.

[18] Hornung A. and Kobbelt L. (2006), Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information, Eurographics / ACM SIGGRAPH SGP '06 Conference Proceedings, ACM Press.

[19] Kazhdan, M., Bolitho, M. and Hoppe H. (2006), Poisson surface reconstruction, Eurographics/ACM SIGGRAPH SGP '06 Conference Proceedings, 61-70, ACM Press.

[20] Kutulakos K. N. and Seitz, S. M. (2000), A theory of shape by space carving, *Int. J. Comput. Vision* **38**(3), 199-218, 2000.

[21] Laurentini, A. (1994), The visual hull concept for silhouette based image understanding. *IEEE PAMI* 16(2), 150-162, 1994.

[22] Lorensen, W. E. and Cline, H. E. (1987), Marching cubes: A high resolution 3d surface construction algorithm, ACM SIGGRAPH '87 Conference Proceedings, 163-169, ACM Press.

[23] Mederos, B., Amenta, N., Velho, L. and de Figueiredo, L. H. (2005), Surface reconstruction from noisy point clouds, Eurographics/ACM SIGGRAPH SGP '05 Conference Proceedings, 53-62. ACM Press.

[24] Montani, C., Scateni, R. and Scopigno, R. (1994), A modified look-up table for implicit disambiguation of marching cubes, *The Visual Computer*, **10**(6), 353-355.

[25] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G. and Seidel, H.-P. (2003), Multi-level partition of unity implicits, *ACM Trans. Graph.*, **22**(3), 463-470.

[26] Paris, S., Sillion, F. and Quan, L. (2006), A surface reconstruction method using global graph cut optimization, *International Journal of Computer Vision*, **66**(2), 141-161.

[27] Petitjean, S. and Boyer, E. (2001), Regular and non-regular point sets: Properties and reconstruction, *Computational Geometry – Theory and Application*, **19**(2-3), 101-126.

[28] Pottmann, H., Leopoldseder, S., and Hofer M. (2002), Approximation with active b-spline curves and surfaces, Pacific Graphics '02 Conference Proceedings, 8-25.

[29] Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L. and Cohen-Or, D. (2006), Competing fronts for coarse-to-fine surface reconstruction, Eurographics/ACM SIGGRAPH SGP '06 Conference Proceedings, 389-398, ACM Press.

[30] Sorkine, O. (2005), Laplacian mesh processing, Eurographics 2005 STAR report.

Hongxin Zhang Dr. Hongxin Zhang is an associate professor of the state key laboratory of CAD&CG at Zhejiang University, P.R.China. He received BS. and Ph.D degrees in applied mathematics from Zhejiang University. His research interests include geometric modeling, texture synthesis and machine learning.

Wei Hua He got Ph.D in applied mathematics from Zhejiang University in 2002. He joined the CAD&CG State Key Lab in 2002. His main interests include real-time simulation and rendering, geometry modeling, virtual reality and software engineering.

Hua Liu Dr. Hua Liu is now a senior software engineer at VIA Ltd., Shang Hai. He received Ph.D degree in the State Key Lab. of CAD&CG, Zhejiang University in 2007. His major research interests include geometric modeling and graphics hardware.

Hujun Bao Dr. Hujun Bao received his Bachelor and Ph.D in applied mathematics from Zhejiang University in 1987 and 1993. His research interests include modeling and rendering techniques for large scale of virtual environments and their applications. He is currently the director of State Key Laboratory of CAD&CG of Zhejiang University. He is also the principal investigator of the virtual reality project sponsored by Ministry of Science and Technology of China.

Hongxin Zhang

Hua Liu

Wei Hua

Hujun Bao