Direct reconstruction of displaced subdivision surface from unorganized points

Won Ki Jeong
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
jeong@mpi-sb.mpg.de

Chang Hun Kim
Dept. of Computer Science and Engineering
Korea University, 5-1 Anam-Dong
Sungbuk-Gu, Seoul Korea
chkim@korea.ac.kr

Abstract

In this paper we propose a new mesh reconstruction algorithm that produces a displaced subdivision mesh directly from unorganized points. The displaced subdivision surface is a new mesh representation that defines a detailed mesh with a displacement map over a smooth domain surface. This mesh representation has several benefits — compact mesh size, piecewise regular connectivity — to overcome limitations of irregular mesh produced by ordinary mesh reconstruction scheme, but the original displaced subdivision surface generation algorithm needs an explicit polygonal mesh to be converted. Our approach is producing displaced subdivision surface directly from input points during the mesh reconstruction process. The main ideas of our algorithm are building initial coarse control mesh by the shrink-wrapping like projection and sampling fine surface detail from unorganized points along the each limit vertex normal without any connectivity information of given points. We employ an existing subdivision surface fitting scheme to generate a parametric domain surface, and suggest a surface detail sampling scheme that determines a valid sampling triangle which can be made with combinations of input points. We show several reconstruction examples and applications to show the validity of suggested sampling technique and benefits of the result like multiresolution modeling.

1 Introduction

By the improvement of optical and mechanical technology and the need of realistic modeling, acquiring accurate surface information from a real object has become commonplace. Such technologies include laser scanner, mechanical probe, and structured light give an output in the form of an unorganized points cloud. To be adapted to existing computer graphics system, unorganized points should be converted into a smooth surface or a polygonal mesh. Hence, there is a large literature on mesh reconstruction algorithms. Mesh reconstruction algorithm gives a dense, seamless irregular polygonal mesh as an output. Such meshes are appropriate for expressing fine surface detail, but those are notorious for their huge amount of data. So, many optimization algorithms — simplification, multiresolution, compression, etc. — have been developed.

Displaced subdivision surface, proposed by Lee et al. [10], is the new mesh representation that expresses a detailed model as a scalar displacement map over a smooth subdivision surface. This representation dramatically reduces the amount of data since it requires only a scalar value for expressing a 3D vertex. This can be thought as a lossy-compression since displaced subdivision surface is an approximation of the original mesh, not exact the same mesh. Besides this, parameterization and smoothness of domain surface is automatically defined by a stationary subdivision scheme, and this representation can be converted easily into a bump map to improve rendering performance. By these benefits, displaced subdivision surface can be used as a new mesh structure to overcome the limitation of an irregular mesh produced by existing mesh reconstruction algorithm. But until now, the displaced subdivision surface can only be produced by the mesh conversion process. It means that we need a two-step construction process to get a displaced subdivision surface from unorganized points — mesh reconstruction process and mesh conversion process — in the original displaced subdivision surface conversion pipeline.

Hence, we suggest a new mesh reconstruction algorithm that produces a displaced subdivision surface directly from an unorganized points. The main idea of our algorithm is sampling fine surface detail from unorganized points along the each limit vertex normal directions, not from an explicit polygonal mesh that has connectivity information. Determination of accurate intersection position between the sampling ray and a virtual surface that is inferred by the input points is the most challenge thing. Since our method produces a displaced subdivision surface directly from a range of points, this can avoid post-processing — like mesh re-

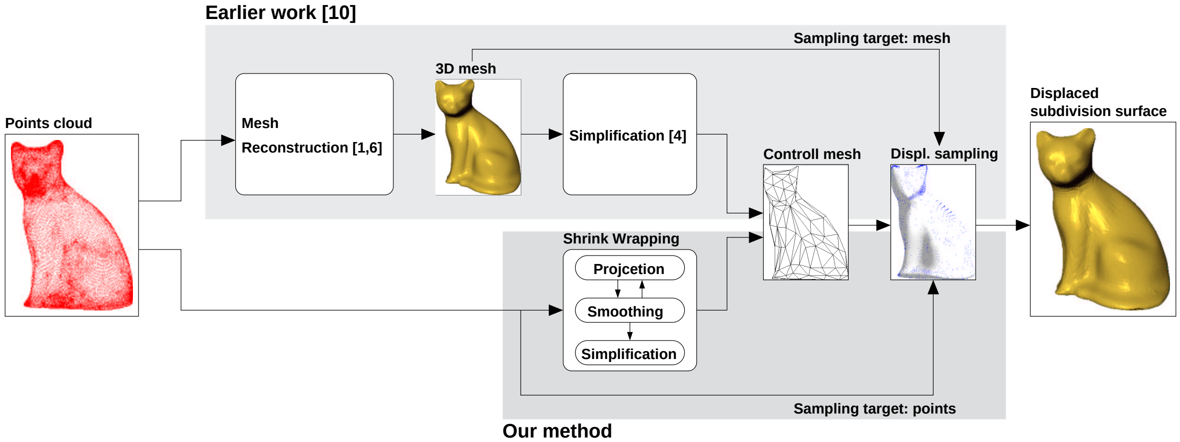


Figure 1. Comparison between the process of existing displaced subdivision mesh conversion algorithm (upper part) & our method (lower part). Our scheme simply produces control mesh directly from points cloud, so we don't need ordinary mesh reconstruction process for post-processing.

construction process — in overall process (Fig.1).

To generate initial control mesh, we start from a bounding cube and successively subdivide, smooth, and project it to given points cloud in shrink-wrapping manner. To capture fine surface detail accurately, we have to deform the control mesh by the subdivision surface fitting scheme [13] to make parameteric domain surface fits well to input points cloud. For each vertex normal of domain surface, we find a proper intersecting triangle that is a combination of input points. This is done by locally and will lessen computational costs that occur in the global energy optimization like the method of Hoppe et al. [5].

The most benefit is the ideal underlying mesh structure that is produced by our algorithm. Output of our algorithm has piecewise-regular structure that is created by successive applying of subdivision process. Since this mesh has subdivision connectivity, we can make multiresolution mesh directly without remeshing process. Larger dataset can be reconstructed and manipulated efficiently since the output of our algorithm has memory-efficient structure.

1.1 Previous Work

1.1.1 Mesh reconstruction

There's a large literature on 3D reconstruction from unorganized points in the computer vision and graphics fields. Most reconstruction schemes have focused on the approximation of a smooth parameteric surface to given points or the derivation of a zero-set of an implicit function [2]. Recently, the extracting triangular mesh from a given set of points is studied. Hoppe et al. [5, 6, 7] proposed the arbitrary 3D mesh reconstruction from unorganized points. He introduced volume-based reconstruction with optimization of energy functions. He employed modified Loop subdivision scheme to optimize result. This method has several advantages — the ability to reconstruct arbitrary topological mesh, optimal and robust result — but it needs much computation. Suzuki et al. [13] proposed a subdivision surface fitting algorithm that uses the limit surface property of approximating subdivision scheme. He changes the shape of control mesh at every level of subdivision to maximally fit limit surface into input points. This method requires small computation, but the result mesh lacks of fine surface detail. Amenta et al. [1] suggested medial-axis and voronoi based surface reconstruction algorithm. It is an interpolating method — it means the vertices on the result mesh are placed on one of input points. It is robust and gives adaptive resolution, but it requires delaunay triangulation operation that is somewhat expensive.

1.1.2 Displacement map

Recenlty, several algorithms that convert an arbitrary mesh into fitted smooth surfaces and displacements are proposed. Main benefit of this work is that the displacement map can be easily transformed into bump map to enhance rendering process. Krishnamurthi et al. [9] proposed a method of smooth surface fitting to a polygonal mesh. They manually divide the input mesh into several sections, and fit B-spline surfaces to them. After fitting process, they sample fine surface detail with displacement vectors. This method can give good smooth surface fitting, but displacements are three-dimensional vectors. Moreover, this method needs a lot of manual process for dividing the input mesh. Recently, Lee et al. [10] proposed another displacement sampling algorithm that uses a subdivision scheme to produce a smooth parametric surface and a scalar displacement value for sampling surface detail.

1.2 Contribution

This algorithm has several benefits as follows:

1. We suggest a new displaced subdivision surface generation technique from unorganized points cloud directly.
2. Our method can skip irregular mesh reconstruction from points cloud in overall process because we can sample fine surface detail from points cloud directly without a target mesh for sampling.
3. We avoid a time-consuming global energy optimization by employing a local subdivision surface fitting and geometric sampling technique, so we can generate a good quality displaced subdivision surface in a short time.

1.3 Algorithm Overview

Our algorithm consists of three steps. First, we make a control mesh from unorganized points using a shrink wrapping process. After that, we subdivide the control mesh several times to get a parametric domain surface that is a base mesh of surface detail sampling. We employ a subdivision surface fitting scheme to prevent shrinkage induced by the subdivision scheme. With given domain surface, we sample surface detail from unorganized points along the each vertex normal direction. Final result is a coarse initial control mesh and scalar displacement values.

2 Problem Definition

Let M be a real world object and $X = \{x_0, x_1, ..., x_n\}$ be a cloud of points that is sampled from M. Then mesh reconstruction algorithm is a process that produces a polygonal mesh M' from X such that $M \approx M'$.

2.1 Input Data

The input X of our algorithm is a cloud of points without connectivity information. This can be acquired from a sampling hardware, e.g. laser range scanner or structured light scanner. We assume that the input points are already registered in a single coordinator system and free from noises. We restrict a topology of reconstructed mesh M' to be a genus-0 spherical topology.

2.2 Parametric Domain and Displacement

Parametric domain $P = \{p_0, p_1, ..., p_k\}$ where displacement sampling is performed is defined as follows:

$$P = S^k m$$

m is a control mesh made with input points, S is a subdivision operator, and k is the subdivision level. Displacement d_i is the distance along a vertex normal of p_i between a vertex and the real world object M that is approximated by input points X (Fig.2).

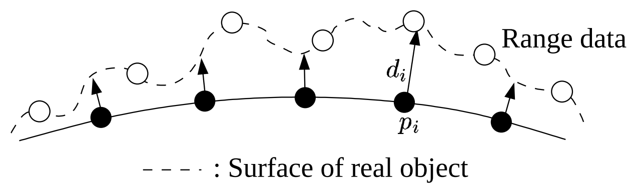


Figure 2. Displacement sampling (2D)

2.3 Mesh Reconstruction

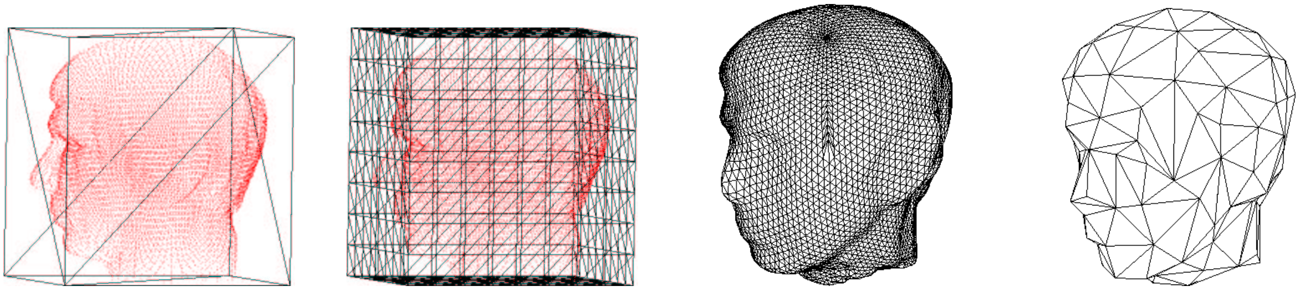
The output of our algorithm is a displaced subdivision mesh. This mesh consists of two parts; a coarse control mesh and a displacement map. Let m be a coarse control mesh, D be a displacement map, and S be a subdivision operator. Then displaced subdivision mesh M' is acquired as follows.

$$M' = S^k m + D$$

k is the subdivision level when the displacement map D was sampled. Since subdivision operator S is known and not needed to be stored in the mesh format, we only have to store a coarse mesh m and a displacement map D which is based on scalar displacements. This causes memory-efficiency of the displaced subdivision mesh format.

3 Domain Surface Generation

A displaced subdivision surface consists of a coarse control mesh with scalar displacement values. Displacement values are captured from a parameteric domain surface which is made by subdivision of the coarse control mesh. In this section, we introduce our initial control mesh generation algorithm from points cloud and the subdivision surface fitting scheme that is used to make parameteric domain surface.



(a) bounding cube (b) subdivided cube (c) shrink wrapped cube (d) after simplification

Figure 3. Control mesh generation by shrink wrapping approach

3.1 Control Mesh Generation

Control mesh generation is the critical process in our algorithm since a control mesh defines the shape of parameteric domain surface and finally it affects the surface detail sampling quality. A control mesh should be both coarse and maximally approximated to input points. In this paper, we employed shrink wrapping approach and mesh simplification to generate initial control mesh of given points cloud.

3.1.1 Shrink Wrapping Approach

The idea of shrink wrapping remeshing has already been given in [8]. Our approach shares the similar idea but is much different from that because our input is just points cloud so that we cannot project it to a common base domain like a sphere in [8]. The main reason that we call our approach shrink wrapping is that we shrink a bounding cube so that it completely wraps up the input points cloud.

First, we make a bounding cube of given points cloud (Fig.3 (a)). Then we linearly (new vertices are placed at the middle of each edges) subdivide it several times until we get a proper high resolution (Fig.3 (b)). To get a high resolution in this stage is important because if we have coarse resolution it is likely to miss some concave or convex region, for example the nose of spock model. To simulate the shrink wrapping process, we repeatedly apply two basic operations — *projection* and *smoothing* operations.

The projection operation is applying the attracting force to each vertices that is a vector between the bounding cube B and the given points cloud X. For each vertices b_i of the bounding cube, we simply find a nearest point x_i, calculate a attracting force vector f_{x_i}, and apply it back to the vertex b_i with weight μ between 0.0 to 1.0.

$$f_{x_i} = b_i - x_i$$

$$B = \sum_i (b_i + \mu f_{x_i})$$

Since the bounding cube is not coarse, there is possibility to share the same point for more than two cube vertices when they calculate attracting forces. In that case, if we apply whole attraction forces to those vertice, several vertices can meet at the same position by this projection operation and it causes non-manifold region as a result. To avoid such undulations, we give a weight μ less than 1.0. Experimentally we chose μ to 0.5.

The smoothing operation is the relaxing of the subdivided bounding cube to acheive uniform sampling. We employ the approximation of Laplacian $\mathcal{L}$ as in [14]. This is the average vector of 1-neighbor edge vectors of a given vertex, and it usually gives shrinkage effects. So we just take the tangential component $\mathcal{L}_t$ (Fig.4 left) of this Laplacian $\mathcal{L}$ which is perpendicular to the vertex normal $\mathbf{n}$. So, the final tangential Laplacian $\mathcal{L}_t$ of a given vertex x_i and iterative smoothing equation is as follows:

$$\mathcal{L}(x_i) = \frac{1}{m} \sum_{j \in 1-nbhd(i)} (x_j - x_i)$$

$$\mathcal{L}_t(x_i) = \mathcal{L}(x_i) - (\mathcal{L}(x_i) \cdot \mathbf{n})\mathbf{n}$$

$$B = B + \lambda \mathcal{L}_t$$

If we apply big λ value (almost near 1.0), we can get uniformly sampled mesh as an output but our shrink wrapping procedure can fail to capture big convex or concave region because mesh can be shrink (there's some shrinkage even though we apply tangential motion only) below such important region and it is hard to be reached by just projecting to the nearest point (Fig.4 right). If we apply small λ value, we cannot get uniformly distributed mesh. We chose 0.2 as the proper λ value by experiments. The example of the result of shrink wrapping process is given in Fig.3 (c).

3.1.2 Coarse Control Mesh Generation

The result of our shrink wrapping process is a high resolution approximation to the given points cloud. However, we only need a very coarse control mesh which approximates given points cloud. So, we just simplify it with the QEM [4] simplification algorithm to get a coarse approximation. An

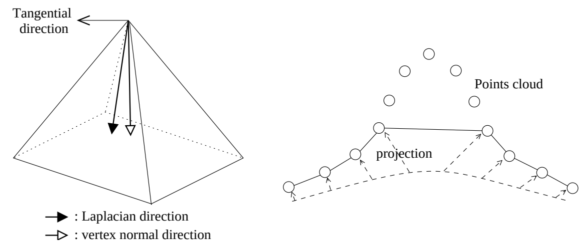


Figure 4. Tangential Laplacian (left) and 2D example of failing of capturing convex region (right)

example of the final coarse control mesh is given in Fig.3 (d).

3.2 Parameteric Domain Surface Generation

This step is to make a smooth parameteric domain surface from a coarse control mesh generated by previous sections. Parametric domain surface is a base mesh for sampling fine surface detail. Since original displaced subdivision surface has employed Loop subdivision scheme [11] to generate a smooth surface, we follow the same subdivision scheme. To successfully capture fine surface detail, a domain surface and input points should be close enough. Since Loop subdivision induces shrinkage, we import subdivision surface fitting scheme [13] to modify control mesh so that the domain surface fits well to input points. In this case, we regard input points as perspective limit position of each vertex of domain surface. We slightly deform the control mesh (Fig.5 (b)) in order to correctly fit domain surface to input points after subdividing it (Fig.5 (c)).

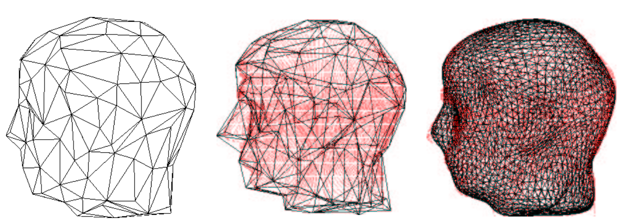


Figure 5. Generating a smooth parameteric domain surface: (a) control mesh with input points, (b) deformed control mesh, (c) parameteric domain surface fitting to input points.

By [11], we find a limit position p_∞ of each vertex p_i on the current domain surface M. Then we find the closest input point q_i of each limit position p_∞. The resultant force r_i is defined as follows:

$$r_i = (q_i - p_i) - \aleph_i \sum_{j \in i^*} (p_j - p_i)$$

where

$$\aleph_i = (\tfrac{3}{8\beta} + k)^{-1}, k = val(p_i)$$

$$\beta = \begin{cases} \frac{3}{16} & \text{for } k = 3 \\ \frac{1}{k}(\frac{5}{8} - (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k})^2) & \text{for } k > 3 \end{cases}$$

We can modify each vertex p_i of the control mesh by the following iterative approximation method.

$$p_i' = p_i + \lambda r_i$$

Optimal value is 0.8 by experiments in [13]. After this fitting process, we subdivide the control mesh until we get the smooth domain surface that has similar complexity of input points by comparing the number of total vertices and send all vertices to their limit positions.

4 Surface Detail Sampling

In the original work [10], Lee et al. seek to compute the signed distance from the limit point of each vertex on the domain surface to the original surface along the vertex normal. Since the sampling target is polygonal mesh, it is easy to find an intersection point between the normal and the original surface. But in our algorithm, we don't have any face or connectivity information. Hence, we find a proper face intersecting with sampling normal vector among combinations of input points.

We define the *valid triangle* T_{valid} as a triangle defined by three input points, which intersects with a given sampling ray and has minimum area. Hence, for given vertex p_i, a displacement value d_i is a distance from p_i to a valid triangle T_{valid} of the sampling ray defined by the vertex normal direction of p_i. Figure 6 shows the valid triangle and proper displacement value d_i of given parametric vertex p_i.

To get a displacement d_i, we have to find T_{valid} for each parametric vertex p_i as follows. First, we find the three points closest to each normal vector. Then we test whether the normal vector intersects the triangle made with these three points or not. If it does, this triangle is valid triangle, and we calculate the signed distance from the domain surface to this triangle along the normal. If not, we add the next near vertex and make combinations to generate other triangular faces until we get an intersected face. If we get multiple intersected faces, we choose the one that has the smallest area among candidate valid triangles. The reason we use the smallest area as the standard of validity is because we want to avoid wrong triangulation as in figure 7.

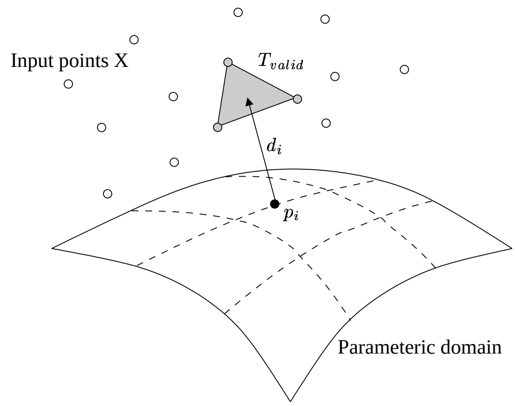


Figure 6. Valid triangle T_{valid} and displacement value d_i

This figure shows 2D concept of selecting a correct valid triangle face. In figure 7, there are two possible valid triangles - (a,b) and (c,d) - since both triangle intersects a given sampling ray. But triangle (a,b) that has the smallest area is the correct valid triangle intuitively. This means that the sampling process is usually performed in the locally almost flat area - we assume that the domain surface is close enough to input points - and in that case the smallest triangle can be the proper choice.

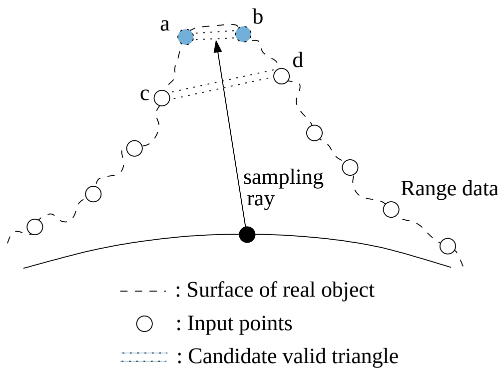
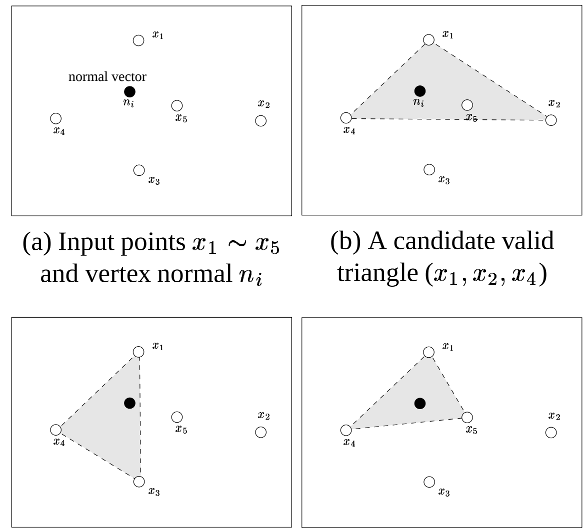


Figure 7. Select valid triangle in two possibilities

Figure 8 shows the example of finding valid triangle T_{valid}. Black dot in this figure is vertex normal vector n_i, which is assigned vertically to this paper. This paper can be regarded as a plane that pass through a parametric vertex p_i and has a normal vector parallel to n_i. White dots are five input points $X = \{x_1, x_2, .., x_5\}$ projected to this plane. There are 10 prospective valid triangles made with five input points (${}_5C_3 = 10$). In these triangles, intersecting triangles with a normal n_i are (x_1, x_2, x_4), (x_1, x_3, x_4), and (x_1, x_4, x_5) (Fig.8 (b)∼(d)), and (x_1, x_4, x_5) is the smallest triangle among them. Hence, this can be a valid triangle to sample displacement value (Fig.8 (d)).



(a) Input points $x_1 \sim x_5$ and vertex normal n_i

(b) A candidate valid triangle (x_1, x_2, x_4)

(c) A candidate valid triangle (x_1, x_3, x_4)

(d) A valid triangle (x_1, x_4, x_5)

Figure 8. Example of selecting a valid triangle

Since there's no connectivity information of input points cloud, we employed octree based spartial search to find the nearest point for given parametric vertex in $O(\log n)$ complexity (n is the number of input points). Octree is constructed by subdividing intial bounding cube recursively.

5. Results

We've implemented our algorithm on Pentium III 866 PC with 384M-RAM. OpenGL, Visual C++, and MFC library were used. The results are shown in the figure 10, 11 at the end of this paper. Table 1 shows the size of testing data and execution time.

	Model	
	Cat	Spock
Input #V	5842	16386
Domain #V	6402	19202
Domain #F	12800	38400
Ctrl mesh	7 sec	1 min 6 sec
Sampling	17 sec	1 min 41 sec
Total	24 sec	2 min 47 sec

Table 1. Testing data and time

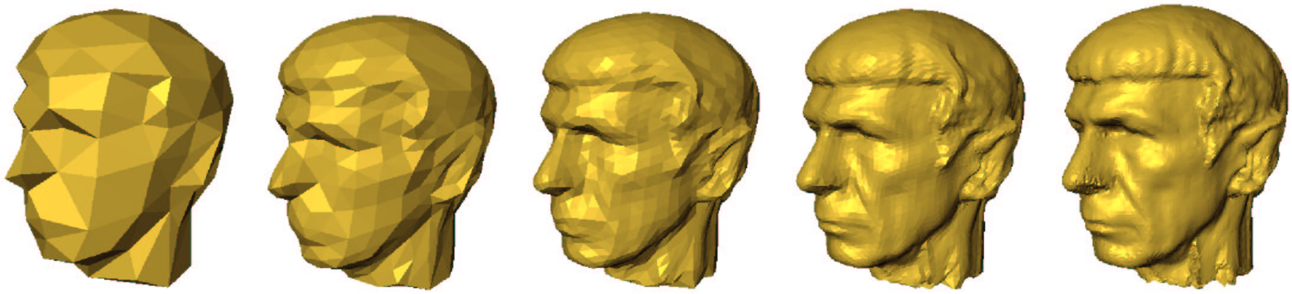


Figure 9. Example of LOD generation (level 0 $\sim$ level 4)

6 Discussion

6.1 Computation time & surface detail

Our algorithm has several benefits compare to existing mesh reconstruction algorithm [5, 13]. Our algorithm reduced reconstruction time compare to global energy optimization based methods. Suzuki et al. [13] suggested that his method is much faster than [5] since he calculated limit positions only with 1-neighbor information and get optimized result by iterative method. But his method lacks of surface detail. Our method can be slower than [13] by additional sampling process, but capture fine surface detail very well. We don't produce an accurate result like time-consuming optimization method, but we can still produce a reasonable approximating result in a short time (less than several minutes). Figure 12 shows the comparison of fine surface detail between the result of Suzuki et al. and ours.

As you can see in the Figure 12, our result shows better fine surface detail near the nose and side hair ((a), red circles). Moreover, Suzuki et al. method can fail to capture important feature like the tip of nose (Fig.12 (b)) if the initial control mesh is not defined manually. Table 2 shows gap errors [13] between input range data and reconstructed 3D mesh. Our method shows far better gap errors since ours can capture fine surface detail as well as overall shape very well.

	Max Error	Min Error	Mean Error
Suzuki et al.	0.123361	0.000108	0.010948
Our method	0.026467	0.000131	0.007379

Table 2. Gap error comparison (spock model)

Our algorithm does not produce accurate result like optimization algorithm of Hoppe et al. but it does produce reasonable detailed surface in a short time. So, our algorithm is useful in ordinary graphics data that does not require high accuracy like CAD model.

6.2 Multiresolution analysis

For multiresolution modeling based on the wavelet function, an arbitrary mesh must be converted into a regular structured mesh [12]. This conversion process is called remeshing [3]. Most existing remeshing schemes use 1–4 subdivision to make a regular structure. Since our algorithm is a sampling process from a smooth subdivision surface to input points, underlying surface structure is piecewise regular - this means the result mesh has subdivision connectivity. So, we can apply our result mesh to multiresolution analysis without remeshing. Figure 9 shows the example of multiresolution representation of our result mesh.

Hoppe et al. & Suzuki et al. methods also produce subdivision mesh as a result, but Hoppe et al. adopted a subdivision scheme to smooth optimized control mesh. So, new vertices created by subdivision do not represent the surface detail. Suzuki et al. method also produces a subdivision mesh, but his method is a kind of surface fitting rather than mesh reconstruction. This method produces a result lacks of fine surface detail as we mentioned earlier, so that is not proper for multiresolution analysis.

6.3 Efficiency of detail sampling algorithm

In this paper, we suggested a surface detail sampling algorithm from points cloud. We search candidate triangles which are compositions of nearest points. Our method is a heuristic determination method that can be used instead of making local delaunay triangulations or parameteric surface generations. We've checked the efficiency of our sampling algorithm by the number of tested triangels for each vertex to find a valid triangle. For cat model, 71% of verteices can find valid triangles at just first tries. 21% of vertices can find valid triangels at fourth tries. Nearly 97% of vertices can find valid triangles at less than tenth tries. This shows our sampling method has enough efficiency for practical use, but still need more robustness.

6.4 Volumetric approach for control mesh generation

A shrink wrapping approach combined with simplification (chapter 3.1) can generate the best approximated control mesh easily. It needs some iterations to project and smooth the bounding cube but all those calculations are local and it can be done in several minutes at most. However, our shrink wrapping approach can only be applied to points cloud that is sampled from a model which has sphere topology. Moreover, even though the model has sphere topology, it is very hard to generate a control mesh by our shrink wrapping approach when the model has large convex region, like ears of the stanford bunny. We think that in the projecting step we can search the nearest point in the direction of each vertex normal. Another way is using volumetric methods to recognize the topology. We've tried a volumetric method for such a complex model.

We use volumetric method only for generating very coarse initial control mesh. Hence, we don't need to assign signed distance function for every vertex as in [6], but need to make coarse cubes to generate an iso-surface by marching cubes algorithm. Since we use only large cubes to generate coarse control mesh, computation and simplification for generating iso-surface is not so expensive. For bunny model, we've used $16\times16\times16$ volume data and the iso-surface can be generated in a few seconds.

7 Conclusion & Future work

In this paper, we introduced a new mesh reconstruction algorithm that extracts a displaced subdivison mesh directly from unorganized points. We simply construct coarse control mesh by shrink wrapping approach and sample fine surface detail directly from input points cloud. Also we avoid global energy minimization and employ local iterative method to enhance sampling domain surface quality. The result mesh has compact representation and is ready to be applied into the wavelet based multiresolution analysis without remeshing since it samples scalar-value displacement and has subdivision connectivity. The underlying structure and smoothness of the domain surface are simply defined by the stationary subdivision scheme. The result mesh has lots of applications — mesh editing, animation, rendering, etc.

Since we have shown the validity of sampling fine surface detail directly from unorganized points, extension of our algorithm for arbitrary topological model can be easily achieved by employing topology realization process — for example medial axis transform or volumetric approach. Moreover, our scheme can be efficiently extended to solve surface fitting problem for a model with known topology, such as making individual facial model from range points using a generic head model.

References

[1] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 415–421, July 1998.

[2] R. Bolle and B. Vermuri. On three-dimensional surface reconstruction methods. *IEEE Trans. on pattern analysis and machine intelligence*, 13(1):1–13, January 1991.

[3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 173–182, August 1995.

[4] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 209–216, August 1997.

[5] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 295–302, July 1994.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, pages 71–78, July 1992.

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, pages 19–26, August 1993.

[8] L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. In *Computer Graphics Forum, Proceedings of Eurographics '99*, volume 18, 3, pages 119–129, September 1999.

[9] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 313–324, August 1996.

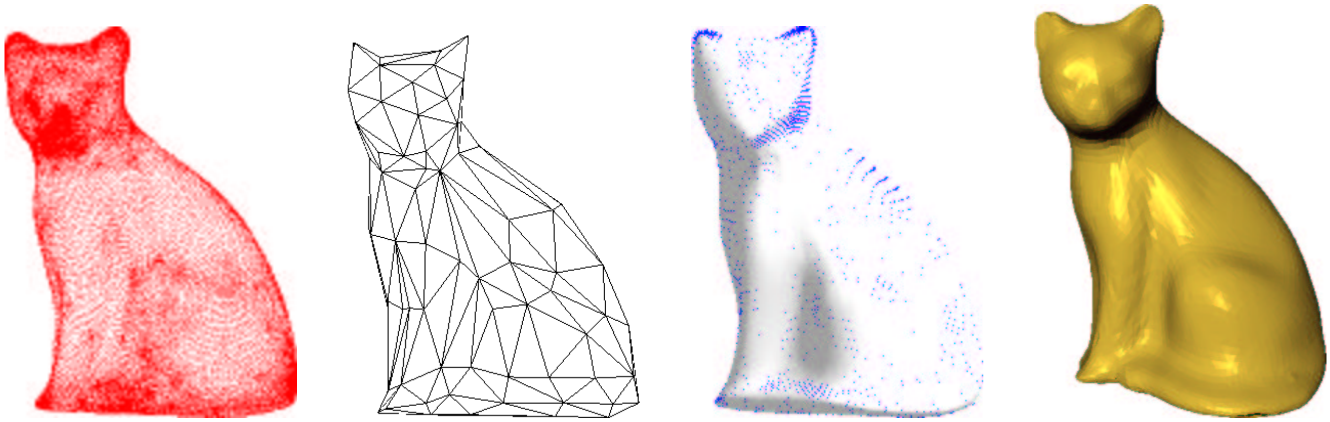
[10] A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Computer Graphics (SIGGRAPH 2000 Proceedings)*, pages 85–94, August 2000.

[11] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah, August 1987.

[12] M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. Technical Report 93-10-05b, Dept. of Computer Science and Engineering, University of Washington, October 1993.

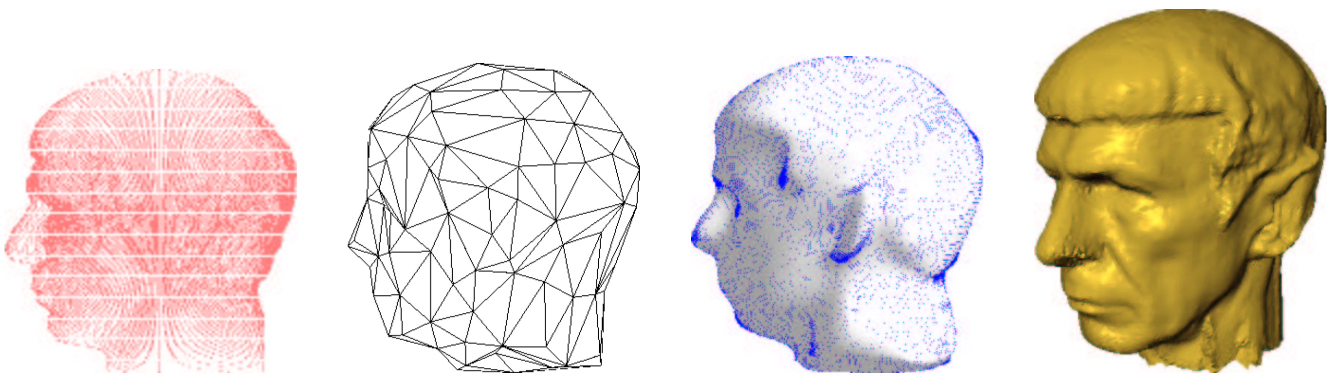
[13] H. Suzuki, S. Takeuchi, and T. Kanai. Subdivision surface fitting to a range of points. In *Proceedings of Pacific Graphics 99*, pages 158–167, 1999.

[14] G. Taubin. A signal processing approach to fair surface design. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 351–358, August 1995.



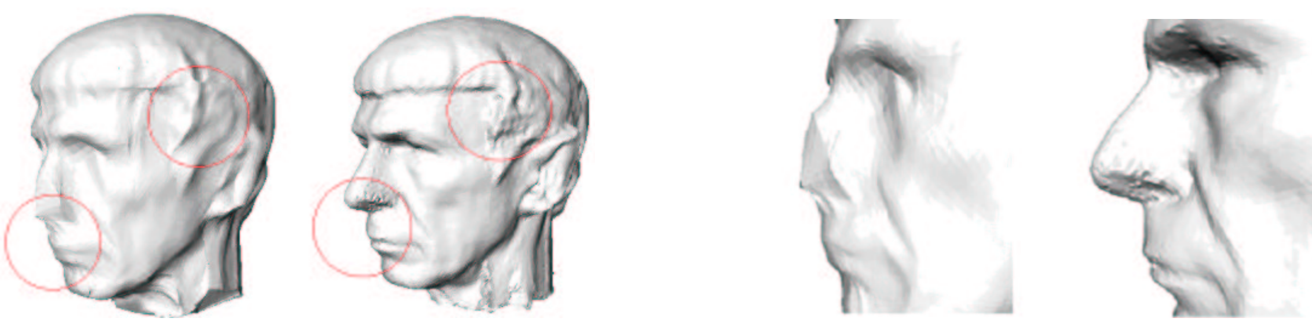
(a) Points cloud (b) Control mesh (c) Dispalcement sampling (d) Result

Figure 10. Reconstruction process and result (cat model)



(a) Points cloud (b) Control mesh (c) Displacement sampling (d) Result

Figure 11. Reconstruction process and result (spock model)



(a) Fine surface detail comparison (b) Zoom in

Figure 12. Comparison of fine surface detail between Suzuki et al. (left) and our method (right)