Smooth Surface Reconstruction using Doo-Sabin Subdivision Surfaces

Fuhua (Frank) Cheng[a], Fengtao Fan[a], Shuhua Lai[b],
Kenjiro T. Miura[c], Conglin Huang[a], Jiaxi Wang[a]

[a]Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA

[b]Department of Mathematics and Computer Science, Virginia State University, Petersburg, VA 23806, USA

[c]Department of Mechanical Engineering, Shizuoka University, Hamamatsu 432-8561, Japan

Abstract. A new technique for the reconstruction of a smooth surface from a set of 3D data points is presented. The reconstructed surface is not represented by a polyhedral approximation, but an everywhere C^1-continuous subdivision surface. The subdivision surface interpolates all the given data points. Besides, the topological structure of the reconstructed surface is exactly the same as that of the data points. Therefore the reconstructed surface is guaranteed to be precise if the data points are taken directly from the sampled object and is good for all subsequent applications. The new technique consists of two major steps. First, an efficient surface reconstruction method is applied to produce a polyhedral approximation to the given data points. A Doo-Sabin subdivision surface that smoothly passes through all the data points in the given data set is then constructed. A new technique is presented for the second step in this paper. The new technique iteratively modifies the vertices of the polyhedral approximation M until a new control mesh $\bar{M}$, whose Doo-Sabin subdivision surface interpolates M, is reached. It is proved that, for any mesh M with any size and any topology, the iterative process is always convergent with Doo-Sabin subdivision scheme. Therefore the surface reconstruction process is well-defined. The new technique has the advantages of both a local method and a global method, and the surface reconstruction process can reproduce special features such as edges and corners faithfully.

Keywords: Surface reconstruction, Loop subdivision surfaces, interpolation

1 Introduction

In many applications, the only available information on a surface is a set of unorganized points sampled from that surface. Before a computation can be perform on that surface, a representation of the surface has to be constructed from the sample points first. This is the problem of surface reconstruction. Problems of this type occur in scientific and engineering applications such as CAD, medical imaging, visualization, computer graphics, computer vision, reverse engineering, etc. The reconstructed surface should be topologically equivalent to and geometrically close to the sampled surface.

Traditional surface reconstruction methods always produce a set of triangles to approximate the surface shape. This usually is not precise enough when small details are needed. One can solve the precision problem by increasing the number of points sampled in the sampling process. This is possible because recent advances in laser technology have made it easier to generate a lot of sample points from the surface of an object. But there are occassions where a discrete representation is not good enough no matter how many points are used in the represenation, such as 3D medical imaging where one needs to scale up an organ or a cross-section frequently. Construction of smooth representation of a surface from unorganized data has been studied for a while and some techniques have already been reported [22]. But the techniques do not guarantee interpolation of the sample points by the generated representation.

In this paper we propose to reconstruct a faithful surface from a set of data points, such that the reconstructed surface is not approximately represented by a polyhedron, but by an everywhere C^1 continuous subdivision surface. The subdivision surface interpolates all the given data points. Besides, the topological structure of the reconstructed surface is exactly the same as that of the data points. Therefore, the representation is guaranteed to be precise if the sampled points are taken directly from the sampled object. This is done in two steps:

- Use an efficient surface reconstruction method to produce a polyhedral approximation to the given sampled points.
- Construct a C^1 continuous subdivision surface to interpolate all the sampled points in the given data set.

While the first step is still a challenging step, it is the second step that is our focus here. Constructing a subdivision surface to interpolate an arbitrary mesh is not a well-solved problem when the number of vertices is large. In this paper we will propose a solution to this poblem.

The remaining part of the paper is arranged as follows. In section 2, background knowledge is reviewed and related work is discussed. The basic idea of the method of surface reconstruction using Doo-Sabin Subdivision Surfaces is presented in Section 3. The correctness and convergence of our method is proven in section 4. Implementation issues and test results are given in Section 5. Section 6 gives some concluding remarks.

2 Background and Related Work

2.1 Surface Reconstruction from Unorganized Points

A number of applications ranging from CAD, computer graphics and mathematical modeling require the reconstruction of a smooth surface from a set of data points. The data points could be densely sampled or sparsely taken from the surface such that they are the representative points of the surface. Many techniques have been proposed to reconstruct an approximated surface from a set of 3D data points. Among them are greedy methods [23], implicit surfaces [21] and Delaunay triangulation, etc. However all of them only lead to a non-smooth polyhedral approximation to the given data points, or to a smooth surface that does not interpolate the input data point set [22]. Therefore without dense sampling of an object surface, none of the methods mentioned above can reconstruct the original surface precisely.

2.2 Subdivision Surfaces

Subdivision surfaces are popular now in Computer Animation, CAD and Geometric Modeling, etc. The ability to model arbitrary topology surfaces makes them more suitable than classical spline surfaces in some applications. The Catmull-Clark subdivision scheme [2] was proposed in 1978, which is the generalization of bicubic spline surface, while the Doo-Sabin subdivision method [1] is the generalization of quadratic spline surface. Later, the Loop subdivision scheme [8] was developed for triangular meshes which generalizes the Box splines. All these three popular subdivision methods are approximating schemes. There are interpolating subdivision schemes that interpolate the given mesh. One of the most famous interpolating subdivision methods is the butterfly subdivision method [6] which was modified subsequently to generate smoother interpolation surfaces in [9]. An interpolating scheme for quadrilateral meshes was proposed in [18].

2.3 Surface Interpolation of Irregular Meshes

Interpolation is a popular technique used in surface design and shape modeling. There are plenty of publications dealing with the interpolation problem using various surface representations. Interpolation methods based on subdivision surfaces have also been developed. One group of them requires solving a global system of linear equations, like [3, 4]. To avoid the computational cost of solving a large system of linear equations, other methods have been developed. In [5], an always-working method solved the problem using a two-phase subdivision method. The method proposed in [15] avoids exactly solving a system of linear equations by using the concept of similarity. The approach presented in [7] avoids solving a system of linear equations by using quasi-interpolation.

In this paper, based on the results obtained from traditional surface reconstruction methods which produce a polyhedral approximation to the given sample points, we present a new iterative interpolation method using Doo-Sabin subdivision surface. Our iterative method is an extension of the *progressive iterative interpolation* method for B-splines [13, 10, 12]. The idea of our iterative interpolation method is to use the differences between the (original) mesh to be interpolated and the Doo-Sabin surface of current mesh to get a new mesh. This iterative process will converge to a Doo-Sabin surface interpolating the original mesh. The updating operation at each level of the iteration is done by a local operation for each vertex in current mesh. Therefore our method possesses the property of a local method. On the other hand, our method has the form of a global method due to its actual global linear effect. Therefore, our method has the advantages of both a local method and a global method. Experimental results demonstrate the efficiency and ability of our method in handling large meshes.

3 Surface Reconstruction using Doo-Sabin Subdivision Surfaces

As mentioned above, there are two major steps in the new surface reconstruction process. First we apply an efficient surface reconstruction method to produce a polyhedral approximation to the given data points, then we find an interpolatory surface for the obtained polyhedral approximation in the first step. There are many efficient approaches that we can use for the first step [21, 22, 23]. In this paper we regard the polyhedral approximation obtained from the first step as the control mesh of a Doo-Sabin subdivision surface and focus on how to construct an interpolating surface for the control mesh.

3.1 Doo-Sabin subdivision surfaces

In Doo-Sabin subdivision scheme, new polygons are built from the old mesh in the following way. An edge point is formed from the midpoint of each edge. A face point is formed as the centroid of each polygon of the mesh. Finally, each vertex in the new mesh is formed as the average of a vertex in the old mesh, a face point for a polygon that is incident to that old vertex, and the edge points for the two edges that belong to that polygon and are adjacent to that old vertex.

The new vertices then are connected. There will be two vertices along each side of each edge in the old mesh, by construction. These pairs are connected, forming quadrilaterals across the old edges. Within each old polygon, there will be as many new vertices as there were vertices in the polygon. These are connected to form a new, smaller, inset polygon. And finally, around each old vertex there is a new vertex in the adjoining corner of each old polygon. These are connected to form a new polygon with as many edges as there were polygons around the old vertex. The new mesh, therefore, will create quadrilaterals for each edge in the old mesh, will create a smaller m-sided polygon for each m-sided polygon in the old mesh, and will create an n-sided polygon for each n-valence vertex. After one application of the scheme all vertices have a valence of four. So subsequent applications will create quadrilaterals for the vertices only. All n-sided polygons are retained in the subdivision process, and shrink to extraordinary points as the subdivision scheme is repeatedly applied.

For a vertex V of valance n (See Fig. 1), if its adjacent edge points are E_i, $1 \le i \le n$ and its adjacent face points are F_j^i, $1 \le i \le n$, $1 \le j \le m_i - 3$, where m_i is the number of edges in the ith

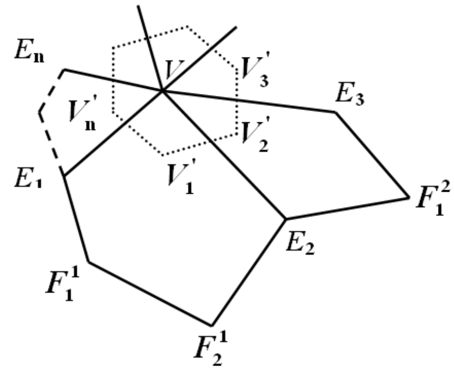


Figure 1: A vertex V of valence n and the new, adjacent vertex points generated after one Doo-Sabin subdivision.

adjacent face, then after one subdivision we have

$$V_i' = (\frac{1}{2} + \frac{1}{4m_i})V + (\frac{1}{8} + \frac{1}{4m_i})E_i + (\frac{1}{8} + \frac{1}{4m_i})E_{i+1} + \frac{1}{4m_i}\sum_{j=1}^{m_i-3} F_j^i,$$

where V_i', $1 \le i \le n$ is one of the newly generated vertex points around vertex V after one subdivision (See Fig. 1).

After each subdivision we have an n-sided polygon around vertex V, which will remain to be n-sided in the subdivision process, and shrink to a limit point as the scheme is repeatedly applied. The limit point corresponding to V on the limit surface can be calculated as follows.

$$\mathbf{V}_\infty = \frac{1}{n}\sum_{i=1}^{n}\mathbf{V}_i'$$

The above formula can be expanded and hence $\mathbf{V}_\infty$ can be more precisely rewritten as follows.

$$\mathbf{V}_\infty = \frac{1}{n}\Big(\sum_{i=1}^{n}\frac{4m_i+2}{8m_i}\mathbf{V} + \sum_{i=1}^{n}\big(\frac{m_i+2}{8m_i} + \frac{m_{i-1}+2}{8m_{i-1}}\big)\mathbf{E}_i + \sum_{i=1}^{n}\big(\sum_{j=1}^{m_i-3}\frac{2}{8m_i}\mathbf{F}_j^i\big)\Big). \qquad (1)$$

3.2 Progressive interpolation using Doo-Sabin subdivision surfaces

For a given a mesh M^0, we will find a new mesh M whose Doo-Sabin limit surface interpolates all vertices of M^0. Instead of solving a global system of linear equations, we develop a progressive iterative method which only locally manipulates vertices of the control mesh by an affine operation at each level of iteration. The iteration process is described as follows.

Initially, for each vertex $\mathbf{V}^0$ of M^0, we compute the difference vector between this vertex and its limit point on the Doo-Sabin surface $\mathbf{S}^0$ calculated from the equation (1),

$$\mathbf{D}^0 = \mathbf{V}^0 - \mathbf{V}^0_\infty,$$

and add the differences $\mathbf{D}^0$ to the vertex $\mathbf{V}^0$.

$$\mathbf{V}^1 = \mathbf{V}^0 + \mathbf{D}^0.$$

Therefore, we get a new control mesh M^1 whose vertices are computed as $\mathbf{V}^1$. By iteratively repeating this process, we get a sequence of control meshes $M^0,\ M^1,\ M^2 \cdots$.

In general, if $\mathbf{V}^k$, $(0 \le k < \infty)$, is the new location of vertex $\mathbf{V}$ after k iterations of the above process and M^k is the control mesh consists of all the new $\mathbf{V}^k$'s, then we denote the Doo-Sabin limit surface of M^k, $\mathbf{S}^k$. We first compute the distance between $\mathbf{V}$ and the limit point $\mathbf{V}^k_\infty$ of $\mathbf{V}^k$ on $\mathbf{S}^k$

$$\mathbf{D}^k = \mathbf{V}^0 - \mathbf{V}^k_\infty. \tag{2}$$

We then add this distance to $\mathbf{V}^k$ to get $\mathbf{V}^{k+1}$ as follows:

$$\mathbf{V}^{k+1} = \mathbf{V}^k + \mathbf{D}^k.$$

The set of new vertices is called M^{k+1}.

This process generates a sequence of control meshes M^k and a sequence of corresponding Doo-Sabin surfaces $\mathbf{S}^k$. $\mathbf{S}^k$ converges to an interpolating surface of M^0 if the distance between $\mathbf{S}^k$ and M^0 converges to zero (i.e., $\mathbf{D}^k \to 0$). Therefore the key task here is to prove that $\mathbf{D}^k$ converges to zero when k tends to infinity.

4 Proof of Convergence

To prove the convergence of the above iterative process, we need a lemma about the eigenvalues of the product of positive definite matrices.

Lemma 1 *Eigenvalues of the product of positive definite matrices are positive.*

The proof of Lemma 1 follows immediately from the fact that if P and Q are square matrices of the same dimension, then PQ and QP have the same eigenvalues (see, e.g., [16], p.14).

As mentioned above, to prove that the iterative interpolation process converges, we must prove that the difference $\mathbf{D}^k$ approaches zero when k tends to infinity. Note that $\mathbf{D}^k$ can be expanded as follows.

$$\begin{aligned}
\mathbf{D}^k &= \mathbf{V}^0 - \mathbf{V}^k_\infty \\
&= \mathbf{V}^0 - \frac{1}{n}\Bigg(\sum_{i=1}^{n}\frac{4m_i+2}{8m_i}\mathbf{V}^k + \sum_{i=1}^{n}\Big(\frac{m_i+2}{8m_i}+\frac{m_{i-1}+2}{8m_{i-1}}\Big)\mathbf{E}_i^k + \sum_{i=1}^{n}\Big(\sum_{j=1}^{m_i-3}\frac{2}{8m_i}(\mathbf{F}_j^i)^k\Big)\Bigg) \\
&= \mathbf{V}^0 - \frac{1}{n}\Bigg(\sum_{i=1}^{n}\frac{4m_i+2}{8m_i}\mathbf{V}^{k-1} + \sum_{i=1}^{n}\Big(\frac{m_i+2}{8m_i}+\frac{m_{i-1}+2}{8m_{i-1}}\Big)\mathbf{E}_i^{k-1} + \sum_{i=1}^{n}\Big(\sum_{j=1}^{m_i-3}\frac{2}{8m_i}(\mathbf{F}_j^i)^{k-1}\Big)\Bigg) \\
&\quad - \frac{1}{n}\Bigg(\sum_{i=1}^{n}\frac{4m_i+2}{8m_i}\mathbf{D}^{k-1} + \sum_{i=1}^{n}\Big(\frac{m_i+2}{8m_i}+\frac{m_{i-1}+2}{8m_{i-1}}\Big)\mathbf{D}^{k-1}_{\mathbf{E}_i} + \sum_{i=1}^{n}\Big(\sum_{j=1}^{m_i-3}\frac{2}{8m_i}\mathbf{D}^{k-1}_{\mathbf{F}_j^i}\Big)\Bigg) \\
&= \mathbf{D}^{k-1} - \frac{1}{n}\Bigg(\sum_{i=1}^{n}\frac{4m_i+2}{8m_i}\mathbf{D}^{k-1} + \sum_{i=1}^{n}\Big(\frac{m_i+2}{8m_i}+\frac{m_{i-1}+2}{8m_{i-1}}\Big)\mathbf{D}^{k-1}_{\mathbf{E}_i} + \sum_{i=1}^{n}\Big(\sum_{j=1}^{m_i-3}\frac{2}{8m_i}\mathbf{D}^{k-1}_{\mathbf{F}_j^i}\Big)\Bigg)
\end{aligned} \tag{3}$$

Eq. (3) can be represented in a compact matrix form as follows.

$$\left[\mathbf{D}_1^k, \mathbf{D}_2^k, \ldots, \mathbf{D}_m^k\right]^T = (I-B)\begin{bmatrix}\mathbf{D}_1^{k-1}\\ \mathbf{D}_2^{k-1}\\ \vdots \\ \mathbf{D}_m^{k-1}\end{bmatrix} = (I-B)^k\begin{bmatrix}\mathbf{D}_1^{0}\\ \mathbf{D}_2^{0}\\ \vdots \\ \mathbf{D}_m^{0}\end{bmatrix}$$

where m is the number of vertices in the given mesh, I is an identity matrix of size $m \times m$, and B is a matrix of the following form:

$$B = \begin{pmatrix} \frac{1}{n_1}(\sum_{i=1}^{n_1} \frac{4m_i+2}{8m_i}) & \cdots & \frac{1}{n_1}(\frac{m_i+2}{8m_i} + \frac{m_{i-1}+2}{8m_{i-1}}) & \cdots & \frac{1}{n_1}(\frac{2}{8m_i}) & \cdots \\ \vdots & \ddots & & & & \\ \frac{1}{n_i}(\frac{m_i+2}{8m_i} + \frac{m_{i-1}+2}{8m_{i-1}}) & & \frac{1}{n_i}(\sum_{i=1}^{n_i} \frac{4m_i+2}{8m_i}) & & & \\ \vdots & & & \ddots & & \\ \frac{1}{n_j}(\frac{2}{8m_i}) & & & & \frac{1}{n_j}(\sum_{i=1}^{n_j} \frac{4m_i+2}{8m_i}) & \\ \vdots & & & & & \ddots \end{pmatrix}$$

Each entry of matrix B can be directly derived from Eq. (1). Now, to prove $\mathbf{D}^k$ approaches zero when k tends to infinity, we just need to show that $(I-B)^k$ approaches zero when k tends to infinity.

Obviously, V^{i+1}, limit points of the mesh control points V^i, ying on the Doo-Sabin subdivision surface S^i, now can be represented in matrix form as $V^{i+1} = BV^i$. Note that B can be decomposed into the product of a diagonal matrix Λ and a symmetric matrix T as follows

$$B = \Lambda T$$

where Λ is of the following form

$$\Lambda = \begin{pmatrix} \frac{1}{n_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{n_2} & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & & & \frac{1}{n_m} \end{pmatrix}$$

and T is of the following form

$$T = \begin{pmatrix} \sum_{i=1}^{n_1} \frac{4m_i+2}{8m_i} & \cdots & (\frac{m_i+2}{8m_i} + \frac{m_{i-1}+2}{8m_{i-1}}) & \cdots & \frac{2}{8m_i} & \cdots \\ \vdots & \ddots & & & & \\ (\frac{m_i+2}{8m_i} + \frac{m_{i-1}+2}{8m_{i-1}}) & & \sum_{i=1}^{n_i} \frac{4m_i+2}{8m_i} & & & \\ \vdots & & & \ddots & & \\ \frac{2}{8m_i} & & & & \sum_{i=1}^{n_j} \frac{4m_i+2}{8m_i} & \\ \vdots & & & & & \ddots \end{pmatrix}$$

Note that if (V_i, V_j) is an edge of a mesh, then (V_j, V_i) is an edge of this mesh as well; if (V_i, V_j) is an edge of a face, then so is (V_j, V_i). In other words, the relationship between two edge vertices or two face vertices is symmetric. It is then easy to see that T is symmetric. Furthermore, it can be proved that matrix T is positive definite.

Proposition 1 *The matrix T is positive definite.*

Proof: It is well-known that a symmetric and strictly diagonally dominant matrix with positive diagonal entries is a positive definite matrix. Because all the coefficients in the Doo-Sabin subdivision process are non-negative, it is easy to check that the diagonal entries of T are positive numbers. Therefore we just need to show that T is a strictly diagonally dominant matrix. According to equation (1), each row of matrix T satisfies

$$\begin{aligned} T_{kk} - \sum_{l=1, l\neq k}^{n_k} T_{lk} &= \sum_{i=1}^{n_k} \frac{4m_i+2}{8m_i} - 2\sum_{i=1}^{n_k} \frac{m_i+2}{8m_i} - \sum_{i=1}^{n_k}(\sum_{j=1}^{m_i-3} \frac{2}{8m_i}) \\ &= \sum_{i=1}^{n_k} \frac{4}{8m_i} > 0 \end{aligned}$$

Hence, T is strictly diagonally dominant and, consequently, T is positive definite.

With the above results, we are ready to prove the convergence of the iterative interpolation process.

Proposition 2 *The iterative interpolation process for Doo-Sabin subdivision surface is convergent.*

Proof: As mentioned above, we just need to prove that $(I - B)^k$ approaches zero when k tends to infinity, where B is defined above and I is an identity matrix. Recall that matrix T is a symmetric positive definite matrix, and so is the diagonal matrix Λ. According to Lemma 1, $B = \Lambda T$, we can conclude that B only has positive eigenvalues. Since Doo-Sabin subdivision scheme satisfies the convex hull property, we have $\|B\|_\infty = 1$, which implies all eigenvalue λ_i of B satisfy $|\lambda_i| \leq 1$. Therefore, all eigenvalues of B satisfy $0 < \lambda_i \leq 1$. Based on this result, it is easy to see that the eigenvalues of matrix $(I - B)$ satisfy $0 \leq 1 - \lambda_i < 1$. Consequently, $(I - B)^k$ approaches zero when k tends to infinity. The convergence of the iterative interpolation process for Doo-Sabin subdivision surfaces then is a direct consequence.

5 Implementation & Results

Implementation of the surface reconstruction technique using Doo-Sabin subdivision surfaces is done on a Windows platform using *OpenGL* as the supporting graphics system. Due to the combination of local and global advantages, the iterative interpolation method is very efficient and can handle very large data sets easily. Besides, our experiment results show that our approach can generate visually pleasing surfaces though there is no fairness parameter in the interpolation scheme.

Table 1: Doo-Sabin surface based progressive interpolation: test results.

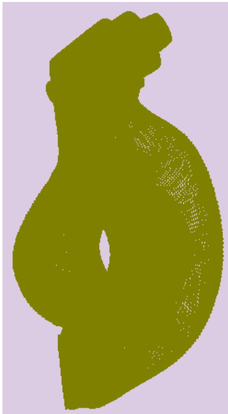
Model	# of data points	# of vertices in poly. Approx.	# of iterations	Error
CubeHC	81920	7666	9	10^{-6}
Goblet	129280	8082	7	10^{-6}
Rockarm	203904	13984	5	10^{-6}
Beethoven	262016	16378	5	10^{-6}

Many examples have been tested and some examples are presented in Fig. 2. In Fig. 2, the input 3D data points for these examples are listed in the first row, the corresponding polyhedral approximations, obtained after applying the surface reconstruction method [23], are listed in the second row, and the reconstructed C^1-continuous Doo-Sabin subdivision surfaces which interpolate the corresponding polyhedral approximations are shown in the third row. We also tabulate some of the testing parameters (see Table 1), such as the number of data points in the input model, the number of vertices in the polyhedral approximation obtained from applying a traditional surface reconstruction method [21, 23], the number of iterations used in the iterative interpolation process to get the interpolating surface and error tolerance used to stop the iteration.

Note that the number of data points in the input 3D model is not the same as the number of vertices in the obtained polyhedral approximation. This is because we made some simplification such that the obtained polyhedral approximations are not as dense as the input data set and meanwhile, without losing much precision (by tolerating a small given error, say 10^{-6}).

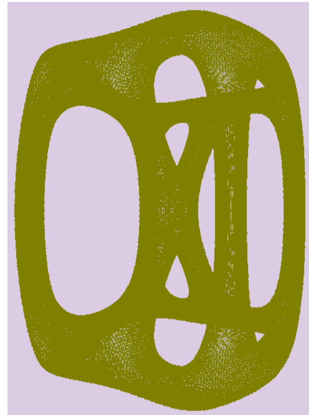
6 Concluding Remarks

A new technique for the reconstruction of a smooth surface from a set of 3D sample points is presented. The reconstructed surface is not represented by a polyhedral approximation, but an

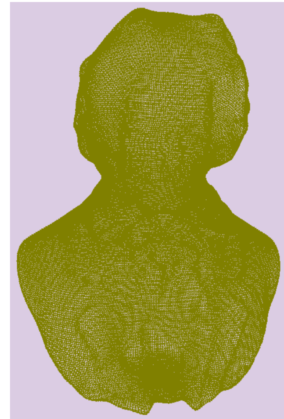


(a) Data Points

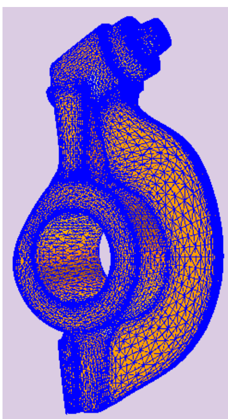
(b) Data Points



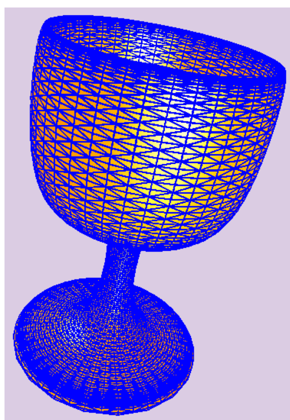
(c) Data Points



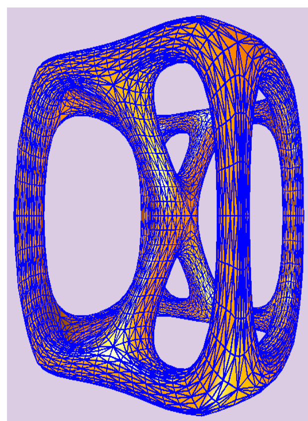
(d) Data Points



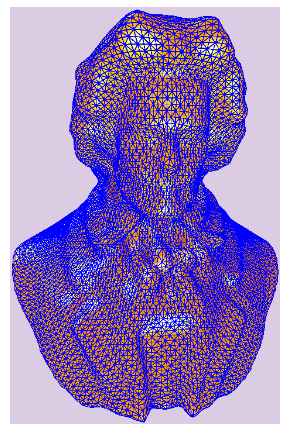
(e) Polyhedral Approximation



(f) Polyhedral Approximation



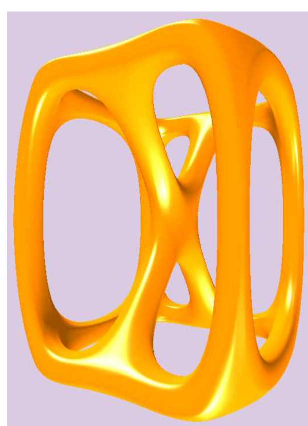
(g) Polyhedral Approximation



(h) Polyhedral Approximation

(i) Reconstruction by Interpolation

(j) Reconstruction by Interpolation



(k) Reconstruction by Interpolation

(l) Reconstruction by Interpolation

Figure 2: Examples of surface reconstruction using Doo-Sabin subdivision surfaces.

everywhere C^1-continuous subdivision surface which interpolates all the sample points. The reconstruction process employs a two-step approach: a surface reconstruction step and a surface interpolation step. The first step produces a polyhedral approximation to the sampled surface from the sample points. The second step produces a Doo-Sabin subdivision surface that interpolates all the sample points. The second step is the focus of this paper. The interpolating surface is generated by iteratively modifying the vertices of the polyhedral approximation M until a control mesh $\bar{M}$, whose Doo-Sabin subdivision surface interpolates M is reached. It is proved that, for any mesh M with any size and any topology, the iterative process is convergent with Doo-Sabin subdivision surfaces. Therefore the surface reconstruction process is well-defined. The new technique has the advantages of both a local method and a global method. Therefore it can handle data set of any size while capable of generating a faithful approximation of the sampled surface no matter how complicated the shape and topolgy of the surface. The surface reconstruction process can also reproduce special features such as edges and corners faithfully.

References

[1] Doo D, Sabin M, Behaviour of recursive division surfaces near extraordinary points, *Computer-Aided Design* 1978, 10(6):356-360.

[2] Catmull E, Clark J, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design* 1978, 10(6):350-355.

[3] Halstead M, Kass M, DeRose T, Efficient, fair interpolation using CatmullCClark surfaces, *Proc.SIGGRAPH 1993*, 47-61.

[4] Nasri AH, Surface interpolation on irregular networks with normal conditions, *Computer Aided Geometric Design* 1991, 8:89-96.

[5] Zheng J, Cai YY, Interpolation over arbitrary topology meshes using a two-phase subdivision scheme, *IEEE Trans. Visualization and Computer Graphics* 2006, 12(3):301-310.

[6] Dyn N, Levin D, Gregory JA, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Trans. Graphics* 1990, 9(2):160-169.

[7] Litke N, Levin A, Schröder P, Fitting Subdivision Surfaces, *Proc. Visualization* 2001, 319-324.

[8] Loop C, Smooth Subdivision Surfaces Based on Triangles, *Master'thesis*, Dept. of Math., Univ. of Utah, 1987.

[9] Zorin D, Schroder P, Sweldens W, Interpolating Subdivision for Meshes with Arbitrary Topology, *Computer Graphics, Ann. Conf. Series*, 1996, 30:189-192.

[10] de Boor C, How does Agee's method work? *Proc. 1979 Army Numerical Analysis and Computers Conference*, ARO Report 79-3, Army Research Office, 299-302.

[11] Lin H, Bao H, Wang G, Totally positive bases and progressive iteration approximation, *Computer & Mathematics with Applications*, 2005, 50:575-58.

[12] Lin H, Wang, G, Dong C, Constructing iterative non-uniform B-spline curve and surface to fit data points, *Science in China*, Series F, 2004, 47(3):315-331.

[13] Qi D, Tian,Z, Zhang Y, Zheng JB, The method of numeric polish in curve fitting, *Acta Mathematica Sinica* 1975, 18:173-184 (in Chinese).

[14] Delgado J, Peña JM, Progressive iterative approximation and bases with the fastest convergence rates, *Computer Aided Geometric Design* 2007, 24(1):10-18.

[15] Lai S, Cheng F, Similarity based Interpolation using Catmull-Clark Subdivision Surfaces, *The Visual Computer* 2006, 22(9):865-873.

[16] Magnus IR, Neudecker H, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, New York, John Wiley & Sons, 1988.

[17] Shilane P, Min P, Kazhdan M, Funkhouser T. The Princeton Shape Benchmark, Shape Modeling Int'l, 2004.

[18] Kobbelt L, Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology, *Comput. Graph. Forum* 1996, 5(3):409-420.

[19] Zheng J, Cai Y, Making Doo-Sabin surface interpolation always work over irregular meshes, *The Visual Computer* 2005, 21(4):242-251.

[20] Gerald Farin, *Curves and Surfaces for CAGD A Practical Guide, 5th edition*, Morgan-Kaufmann, 2001.

[21] H. Hoppe, T. DeRose, T. Duchamp, etc., *Surface reconstruction from unorganized points*, SIGGRAPH 1992, 71-78.

[22] T. K. Dey and J. Sun, *An Adaptive MLS Surface for Reconstruction with Guarantees*, Symposium on Geometry Processing 2005, 43–52.

[23] David Cohen-Steiner, Frank Da, *Greedy Delaunay Based Surface Reconstruction Algorithm*, technical report, ECG-TR-124202-01, http://cgal.inria.fr/Reconstruction.