Reverse Engineering Using a Subdivision Surface Scheme

Anshuman Razdan, Pornchai Mongkolnam, Gerald Farin

E-mails: razdan@asu.edu, pornchai@asu.edu, farin@asu.edu
Partnership for Research in Spatial Modeling (PRISM) and
Department of Computer Science and Engineering
Arizona State University, Tempe, Arizona, USA

Abstract

Subdivision surfaces are finding their way into many Computer Aided Design and Animation packages. Popular choices include Loop, Catmull-Clark, Doo-Sabin etc. Subdivision surfaces have many design advantages over traditional use of NURBs. NURB surfaces always are problematic when multiple patches meet.

Reverse engineering (RE) is associated with the idea of scanning physical objects and representing the resulting dense cloud of points with mathematical surfaces. In RE the goal is to convert the dense point of scanned points into a patchwork of NURB surfaces with most effort going into automating the process. With the emergence of subdivision surfaces as popular modeling tools, it only follows that a similar process be devised for this class of surfaces. Our paper looks at developing one such method for Loop surfaces.

Given a dense triangular mesh, we would like to obtain a control mesh for a Loop subdivision surface which approximates the given mesh. This process benefits subdivision surfaces in animation and manipulation as mentioned in [DeR98] that need speed over accuracy with an ability to manipulate the control mesh and to regenerate the smooth surface quickly. Like subdivision wavelets in a multi-resolution analysis [Lou97], our method can perform level-of-details (LOD) with arbitrary topological meshes useful in applications requiring a fast transfer, less storage, and a fast rendering and interaction.

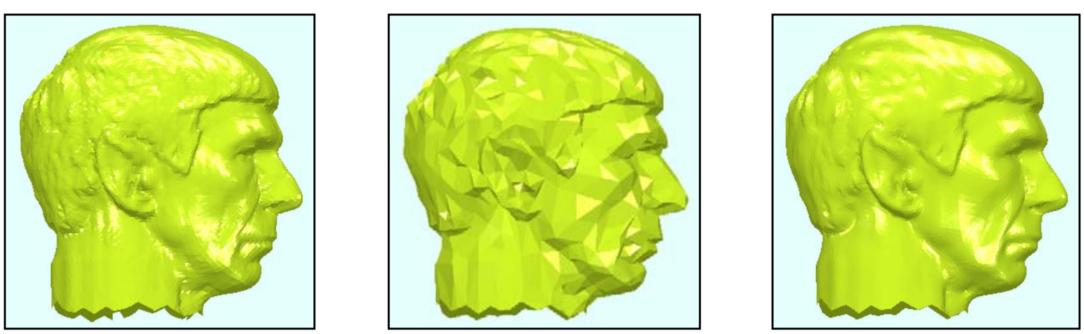
The paper shows the process as well as some early results which are promising. The resulting coarse control mesh is approximately 6.25% of the original mesh therefore this method can also be used as a lossy compression scheme.

1 Introduction

Given dense unorganized data points such as a point cloud from a range scanner, a triangle mesh can be constructed by various methods [Ede94 [Ame98], [Hop92]and [Ber99]. However, triangular meshes obtained are piecewise linear surfaces. For editing, modeling, etc. dense triangle meshes are not optimal solution. Dense triangle meshes with lot of detail are expensive to represent, store, transmit and manipulate. A tensor product NURBS (Non Uniform Rational B-Splines) [Far96] and B-splines are the most popular smooth surface representations. Considerable work has been done on fitting B-spline surfaces to three-dimensional points. This process is often called the Reverse Engineering process wherein a digital representation of the physical object is created. We cite the works of [Loo90], [Ma95], [Eck96], [For88], [For95] and [Kri96] as serious contributions to RE using NURBS/B-splines.

A B-spline surface is a parametric surface, and it needs a parametric domain. [Flo97] and [Flo01] proposed methods for parameterizing a triangular mesh and unorganized points respectively for a single surface patch. A single B-spline patch can only model surfaces with simple topological types such as deformed planar regions, cylinders, and tori. Therefore, it is impossible to use a single non-degenerate B-spline to model general closed surfaces or surfaces with handles. Multiple B-spline patches are needed for arbitrary topological surfaces; however, there are some geometric continuity conditions that must be met for adjacent patches. Therefore, using NURBS/B-splines is not the most desirable approach since it requires high-dimensional constrained optimization. A subdivision surface scheme such as Loops, does not suffer from these problems, has compact storage and simple representation (as triangle base mesh) and can be evaluated on the fly to any resolution with ease. It does not require computation of a domain surface. Hence, it is the focus of this paper.

Lee et al. [Lee00] was the first to unify subdivision surfaces and displacement maps and this was followed by later work [Jeo01]. Our work has been inspired by previous research on multi-resolution analysis [Lou97] and displaced subdivision surfaced [Lee00]. Similar in nature to the subdivision wavelets, we would like to obtain a control mesh approximating original surface with small magnitude of details (as a scalar function) needed to best reconstruct an approximation of the original mesh (Figure 1). The benefit of using a scalar-valued function is that its representation is more compact than its traditional counterpart, a vector-valued geometry representation as used in [Kri96]. One of our contributions is to obtain a control mesh with very small magnitude of displaced values (details) in order to have a very high compression ratio while preserving surface details given a semi-regular (subdivision connectivity) of the original mesh.



(a) original mesh (b) control mesh (c) displaced subdivision surface

Figure 1: Example of an original mesh, its control mesh and its displaced subdivision surface.

2 Related Work

2.1 Subdivision surfaces

A subdivision surface is defined by a refinement of an initial control mesh. In the limit of the refinement process, a smooth surface is obtained. Doo and Sabin [Doo78], and Catmull and Clark [Cat78] first introduced subdivision schemes based on quadrilateral meshes. Their schemes respectively generalized bi-quadratic and bi-cubic tensor product B-splines [Far96]. The triangular based subdivision scheme was introduced by Loop [Loo87] which was a generalization of C^2 quartic triangular B-splines [Far96].

Some authors combined B-splines and subdivision for surface fitting and surface reconstruction. [Tak00] used a Doo-Sabin subdivision scheme [Doo78] to fit a surface to a dense triangular mesh and automatically constructed B-spline surfaces from the subdivision surface. [Ma00] proposed a Catmull-Clark subdivision [Cat78] surface fitting as a network of smoothly connected bi-cubic B-spline surfaces.

Hoppe et al. [Hop94] presented a piecewise smooth surface fitting method to scattered range data points using Loop subdivision scheme [Loo87] to fit the data through an optimization process. The method could model surface of arbitrary topology. The subdivision rules were locally modified to model sharp features such as creases and corners. They used approach by [Hop92] to construct the triangular mesh from the given unorganized points. The number of triangles is reduced and improved by optimizing the energy function as in [Hop93]. Then they employed the energy function optimization to fit a piecewise smooth subdivision surface to the piecewise linear surface obtained from the second phase.

Suzuki et al. [Suz99] used the subdivision limit position (SLP) to repeatedly adjust a control mesh and subdivide it to fit the data points of arbitrary topological objects in their surface fitting method. It quickly captured the geometrical shape of the object because the fitting was made locally at each vertex instead of having to solve a large linear system of equations as done in [Hop94]. It however fails to preserve the sharp features, however.

Lee et al. [Lee00] proposed a new surface representation to an arbitrary triangle mesh, namely a displaced subdivision surface. It generated a detailed surface by displacing a scalar-valued offset over a smooth parametric domain surface instead of using a vector-valued displacement map. To obtain an initial control mesh, a sequence of edge collapse transformation is used to simplify the original mesh. The smooth domain surface is then obtained by applying a number of levels of the Loop subdivision to the initial control mesh, and then at each vertex of this subdivided mesh the SLP (surface limit point) as well as its normal is computed. Finally, the signed distance is computed from the limit point to the original surface along the normal. To get the reconstructed smooth surface to the data points, the control mesh is subdivided to a given level where the displacement are computed, and then the displacement map is added to the subdivided mesh producing the approximated smooth surface.

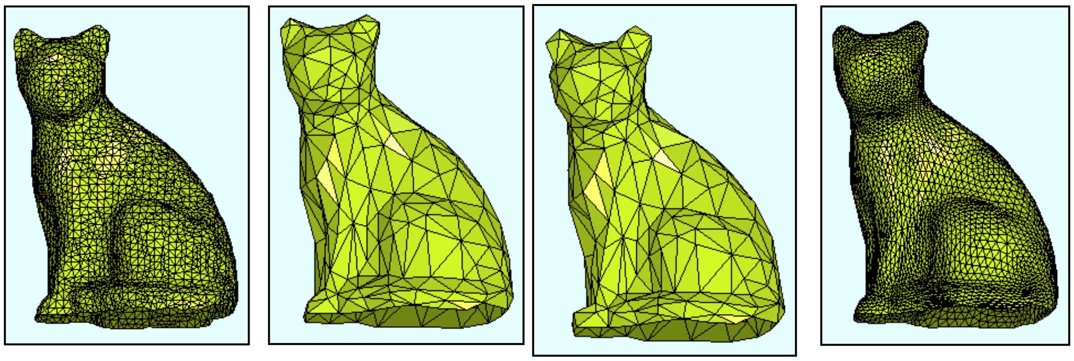
Unlike [Lee00], Jeong et al. [Jeo01] constructed the displaced subdivision surface directly from a point cloud which was homeomorphic to a sphere instead of constructing it from a triangular mesh. The initial control mesh was generated by successively subdividing a bounding cube, smoothing and projecting it to the given point cloud in a shrink-wrapping manner. The reconstructed smooth surface is obtained by adding the displacement to the control mesh subdivided after a certain level.

2.2 Remeshing

Remeshing means to convert a general triangle mesh and into a semi-regular mesh. It is called a semi-regular mesh because most vertices are regular (valence six) except at some vertices

(vertices from a control mesh not having valence six). Remesh algorithms have been proposed by [Eck95], [Lee98], [Kob99], and [Hor01]. Semi-regular mesh can be used for multi-resolution analysis. Through the remeshing process a parameterization can be constructed and used in other contexts such as texture mapping or NURBS patches.

3 Our Work



(a) original (b) simplified (c) adjusted (d) subdivided and displaced

Figure 2: The Remesh process

The overview of our method is as follows: an original mesh is simplified by a quadric error metric (QEM) based on [Gar97] to get a simplified control mesh. We decimate the mesh to 6.25% of the number of original triangles. Based on [Suz99], the control mesh is then adjusted such that after some levels of subdivision the control mesh vertices lie close to the original surface. The adjusted control mesh is then subdivided using the Loop's scheme [Loo87] for two levels obtaining subdivided mesh with its number of triangles close to that of the original mesh. Based on [Lee00], the subdivided mesh vertices are then displaced to the original surface. Our remeshing process is shown in Figure 2. Most or all vertices, however, require displacements during this process even if the control mesh has been adjusted previously. With that in mind, we need to reverse the process of Loop subdivision scheme to obtain a better control mesh with smaller and fewer displacement values (details). A flow chart of our process is shown in Figure 3.

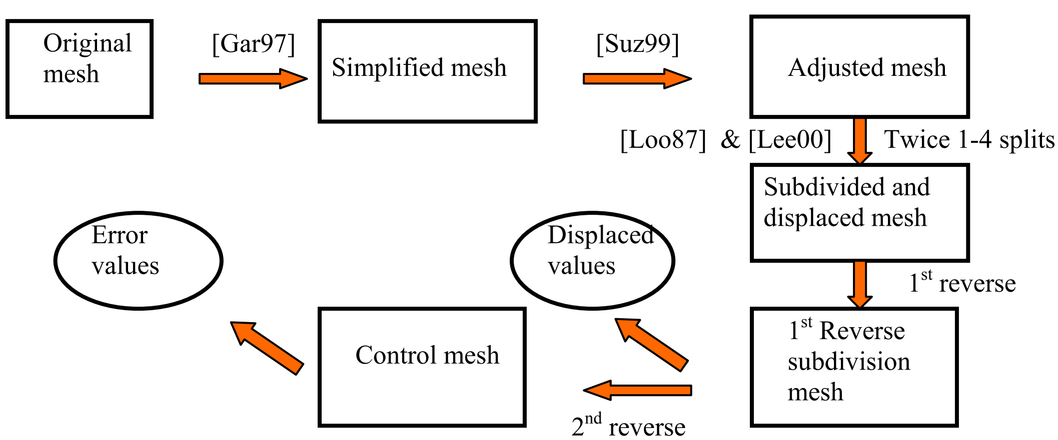


Figure 3: Flow chart of our method.

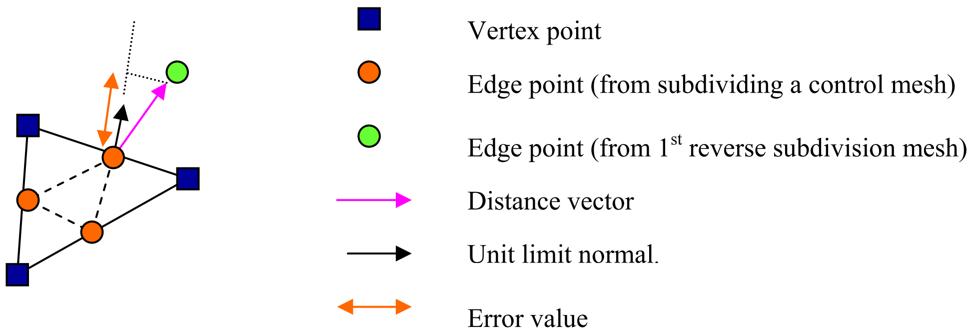


Figure 4: Error value computation for 2nd reverse step.

In the Loop subdivision, each subdividing level produces two types of vertices: *vertex points* and *edge points* as shown in Figure 4. After obtaining the subdivided and displaced mesh, the mesh is semi-regular (subdivision connectivity) and all vertices lie on the original surface.

$$M_{(n+m)xn}\begin{bmatrix} v_1^{i-1} \\ v_2^{i-1} \\ . \\ . \\ . \\ v_{n-1}^{i-1} \\ v_n^{i-1} \end{bmatrix} = \begin{bmatrix} v_1^{i} \\ v_2^{i} \\ . \\ . \\ . \\ v_{n-1}^{i} \\ v_n^{i} \\ e_1^{i} \\ e_2^{i} \\ . \\ . \\ . \\ e_{m-1}^{i} \\ e_m^{i} \end{bmatrix}$$

$$M_{nxn}\begin{bmatrix} v_1^{i-1} \\ v_2^{i-1} \\ . \\ . \\ . \\ v_{n-1}^{i-1} \\ v_n^{i-1} \end{bmatrix} = \begin{bmatrix} v_1^{i} \\ v_2^{i} \\ . \\ . \\ . \\ v_{n-1}^{i} \\ v_n^{i} \end{bmatrix} \qquad \begin{bmatrix} v_1^{i-1} \\ v_2^{i-1} \\ . \\ . \\ . \\ v_{n-1}^{i-1} \\ v_n^{i-1} \end{bmatrix} = M_{nxn}^{-1}\begin{bmatrix} v_1^{i} \\ v_2^{i} \\ . \\ . \\ . \\ v_{n-1}^{i} \\ v_n^{i} \end{bmatrix}$$

Where,
- ***v*** is a vertex point
- ***e*** is an edge point
- ***i*** is subdivision level
- ***M*** is a Loop subdivision matrix

(a) Subdivision matrix (b) Subdivision matrix and its reverse for vertex points only

Figure 5: Reverse subdivision matrices.

The first reverse step applies to the mesh to obtain a coarser 1st reverse subdivision mesh. We don't choose a subdivision matrix in Figure 5(a) because both vertex points and edge points are used in solving linear system of equation. The subdivision matrix is not a square matrix, and we need to do least-square approximation via the normal equations to obtain the control vertices. If that has been done, displaced values would have been required for all vertices (both vertex points and edge points), and by doing that it defeats our goal of having fewer number of required detail values (displaced values and error values) and small magnitudes needed for high compression. Therefore, only the subdivision matrix for vertex points as shown in Figure 5(b) is used instead.

Here, the matrix is a square matrix and is invertible assuming a vertex general position. We could solve for exact control vertices. For large linear system, we use Gauss-Siedel iterative approach to solve the linear system. Now, we need to calculate displaced values and save them for a reconstruction phase. To get them, we internally subdivide the 1st reverse subdivision mesh one level. The vertex points of this subdivided mesh (level 1) are about 25% of total vertices, and they all have zero displaced values. The edge points may or may not require displaced values. The percentage of edge points requires zero displaced values as shown in Table 1. Note that the computation for displaced values for edge points at this level is to find the distance along each vertex limit normal intersecting with the original surface.

In the second reverse step, we again solve for the control vertices of the 1st reverse mesh in similar manner to that of the first reverse step. However, the error values (or what we call the displaced values in the first reverse step) for the edge points are computed differently as shown in Figure 4. The error values are computed as a dot product of edge point unit limit-normal and vector from its position (obtained from subdividing the solved control mesh) to its corresponding position (obtained from the 1st reverse subdivision mesh). The end result is a very small coarse mesh plus some details using which one can approximate the input mesh. The coarse mesh can be used as a model much like the NURBS/B-spline control mesh or the model and detail can be used as a lossy compression scheme.

4 Results

We reconstruct and compare a number of well-known data sets as shown in Figure 6. After obtaining control meshes, we losslessly compress them using software by 3D Compression Technologies Inc. to further reduce the output file size. For the encoding of details (error values and displaced values) we vary the quantization level from 8 to 12 bits per detail value for comparison. Magnitude of Loop vertex limit normal is used as a scaling factor to further reduce the already small detail values. We compare the result between 8-bit and 12-bit detail encoding and found that by encoding with lower bits (8 bits), the reconstructed surface is as good as that of the higher bits (12 bits). In addition, 8-bit quantization would produce higher number zero details, which can be further encoded using variable-length scheme. In the variable-length encoding, each detail value is either encoded by 1 bit for zero detail value or 9 bits otherwise. The ninth bit is to indicate the required detail encoding. As a result, our output surfaces have high fidelity of original surface details as well as high compression ratio. Table 1 shows percentage of vertices requiring no displacedment values at different levels. Comparison of the quantitative compression results among 8-bit detail encoding, 12-bit detail encoding, and 9-bit variable-length detail encoding (OPTZ) are shown in Table 2.

5 Conclusion and Future Work

We have shown that we can approximate any arbitrary topological boundary and non-boundary surfaces with high compression and details. We have combined subdivision surface and scalar-valued displacement for our surface reconstruction. Our domain surface is obtained in such a way that it is close to the original surface, the magnitude of displaced values tends to be very small and is favorable as a compression scheme. With the high compression and faithfully detailed surface, our scheme is promising in areas such as mesh compression, animation, surface editing and manipulation. Future work includes enhancement of the simplification process in order to obtain more regular vertex valences and triangular area of the control mesh and a more sophisticated encoding schemes.

6 Acknowledgements

Authors are partially supported by NSF grant IIS 9980166 at PRISM at Arizona State University. We wish to thank Cyberware for the Horse, Spock and Igea data sets and Stanford University Computer Graphics Laboratory for the Bunny data. We also wish to thank 3D Compression Technologies Inc. for allowing us to use its software to losslessly compress the control meshes. Special thank to Chang Hun Kim and Won Ki Jeong for kindly sending us the Spock point cloud data.

7 References

[Ame98] *Amenta, N., Bern, M., Kamvysselis, M.* A new Voronoi-based surface reconstruction algorithm. Proceedings of the 25th annual conference on Computer graphics and interactive techniques July 1998

[Ber99] *Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G.* The ball-pivoting algorithm for surface reconstruction. Visualization and Computer Graphics, IEEE Transactions on , Volume: 5 Issue: 4 , Oct.-Dec. 1999. Page(s): 349 -359

[Cat78] *Catmull, E., and Clark, J.* Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-aided design 1978.

[DeR98] *DeRose, T., Kass, M., and Truong, T.* Subdivision surfaces in character animation. Proceedings of the 25th annual conference on Computer graphics and interactive techniques, July 1998

[Doo78] *Doo, D., and Sabin, M.* Behavior of recursive division surfaces near extraordinary points. Computer aided design 1978.

[Eck95] *Eck, M., Derose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W.* Multiresolution analysis of arbitrary meshes. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques September 1995

[Eck96] *Eck, M., Hugues Hoppe, H*. Automatic reconstruction of B-spline surfaces of arbitrary topological type. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, August 1996

[Ede94] *Edelsbrunner, H. , Mücke, E. P.* Three-dimensional alpha shapes. ACM Transactions on Graphics (TOG) January 1994. Volume 13 Issue 1

[Far96] *Farin, G.,* Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide. 4th edition. Academic Press, Incorporated, October 1996

[Flo97] *Floater, M*., Parametrization and smooth approximation of surface triangulations, Computer Aided Geometric Design, Volume 14, Issue 3, April 1997, Pages 231-250

[Flo01] *Floater, M., Reimers, M.* Meshless parameterization and surface reconstruction, Computer Aided Geometric Design, Volume 18, Issue 2, March 2001, Pages 77-92

[For88] *Forsey, D. R., and Bartels, R. H.* Hierarchical B-spline refinement. ACM SIGGRAPH Computer Graphics , Proceedings of the 15th annual conference on Computer graphics and interactive techniques June 1988, Volume 22 Issue 4

[For95] *Forsey, D. R., and Bartels, R. H.* Surface fitting with hierarchical splines. ACM Transactions on Graphics (TOG) April 1995, Volume 14 Issue 2

[Gar97] *Garland, M., and Heckbert, P. S.* Surface simplification using quadric error metrics. Proceedings of the 24th annual conference on Computer graphics and interactive techniques August 1997

[Hop92] *Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.* Surface Reconstruction from Unorganized Points. Proceedings of the 19th annual conference on Computer graphics,Pages 71 – 78, 1992

[Hop93] *Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W*. Mesh optimization. Proceedings of the 20th annual conference on Computer graphics, Pages 19 – 26, 1993

[Hop94] *Hoppe, H., DeRose, T., Duchamp, T., and Halstead, M*. Piecewise Smooth Surface Reconstruction. Proceedings of the 21st annual conference on Computer graphics, 1994

[Hor01] *Hormann, K., Labsik, U., and Greiner, G.* Remeshing triangulated surfaces with optimal parameterizations, Computer-Aided Design, Volume 33, Issue 11, 14 September 2001, Pages 779-788

[Jeo01] *Jeong, W. K., and Kin, C. H*. Direct Reconstruction of Displaced Subdivision Surface from Unorganized Points, IEEE, 2001

[Kob99] *Kobbelt, L., Vorsatz, J., Labsik, U., Seidel, H-P.* A Shrink Wrapping Approach to Remeshing Polygonal Surfaces. Computer G raphics Forum 18 (1999), Eurographics '99 issue, pp. C119 -- C130

[Kri96] *Krishnamurthy, V., and Levoy, M.* Fitting smooth surfaces to dense polygon meshes. Proceedings of the 23rd annual conference on Computer graphics and interactive techniques August 1996

[Lee98] *Lee, A., Sweldens, W., Schröder, P., Cowsar, L., and Dobkin, D.* MAPS: multiresolution adaptive parameterization of surfaces. Proceedings of the 25th annual conference on Computer graphics and interactive techniques July 1998

[Lee00] *Lee, A., Moreton, H., Hoppe, H.* Displaced Subdivision Surfaces. Proceedings of the annual conference on Computer graphics (SIGGRAPH 2000), 2000.

[Loo87] *Loop, C.* Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of mathematics, 1987.

[Loo90] *Loop, C., and DeRose, T.* Generalized B-spline surfaces of arbitrary topology. ACM SIGGRAPH Computer Graphics , Proceedings of the 17th annual conference on Computer graphics and interactive techniques September 1990, Volume 24 Issue 4

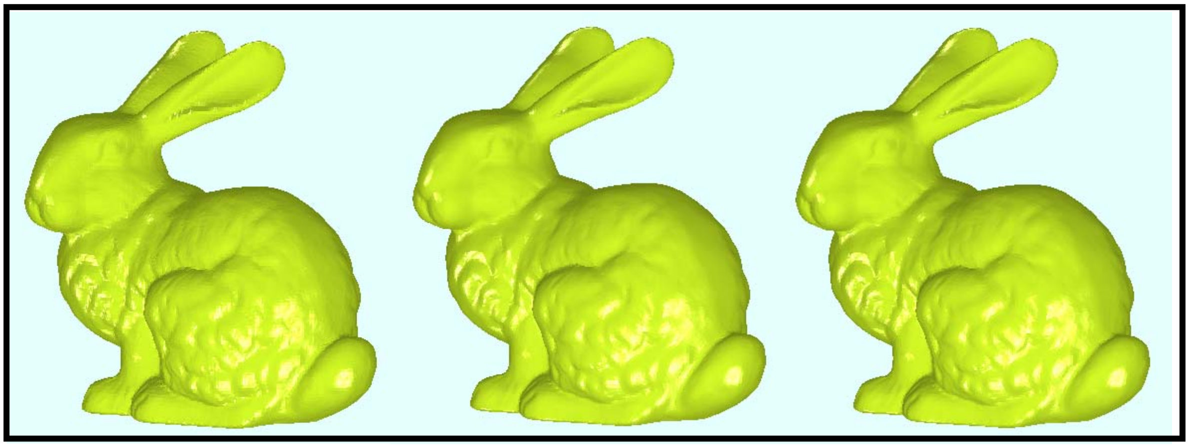
[Lou97] *Lounsbery, M. , DeRose, T.D. , Warren, J.* Multiresolution analysis for surfaces of arbitrary topological type**.** ACM Transactions on Graphics (TOG**).** January 1997. Volume 16, Issue 1

[Ma95] *Ma, W., Kruth, J.P*. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces, Computer-Aided Design, Volume 27, Issue 9, September 1995, Pages 663-675

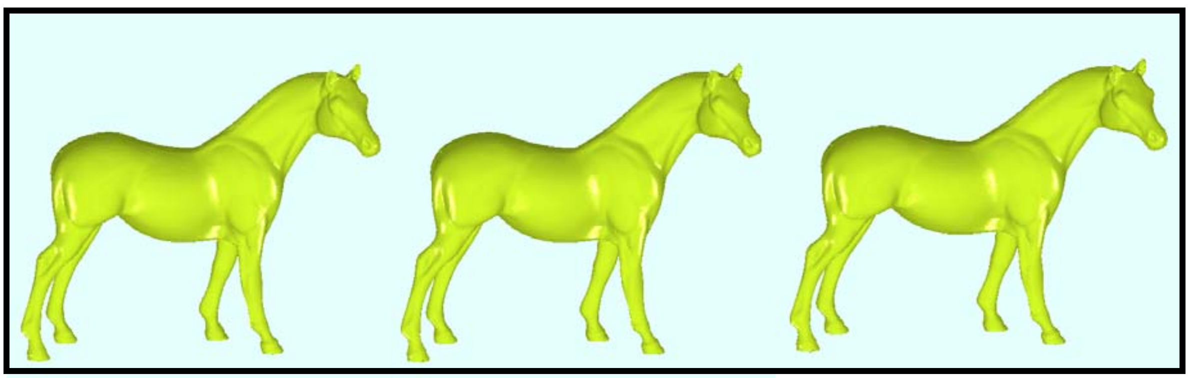
[Ma00] *Ma, W., Zhao, N.* Catmull-Clark surface fitting for reverse engineering applications **.** Geometric Modeling and Processing 2000, Theory and Applications. Proceedings, 2000. Page(s): 274 -283

[Suz99] *Suzuki, H., Takeuchi, S., and Kanai, T.* Subdivision Surface Fitting to a Range of Points. Computer Graphics and Applications, Proceedings. Seventh Pacific Conference, 1999

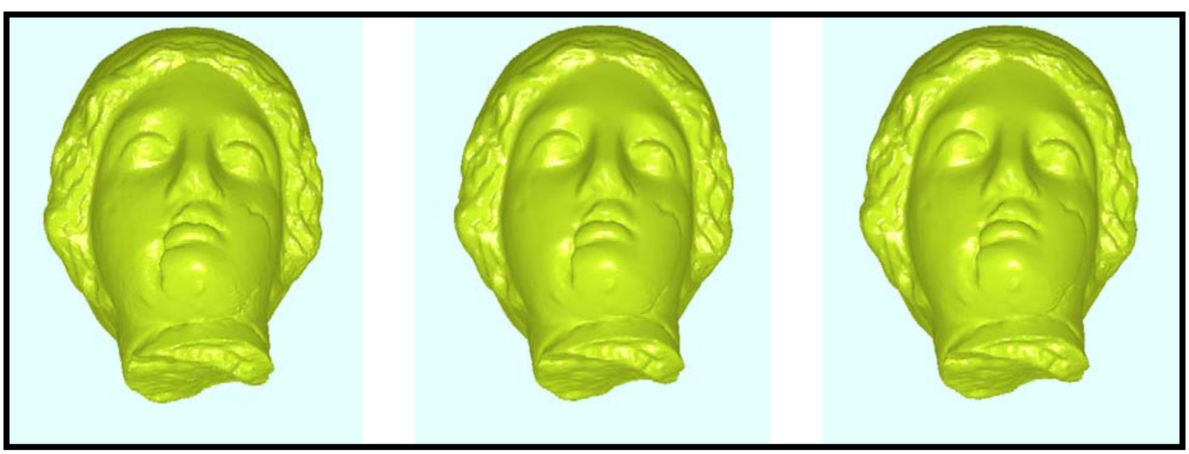
[Tak00] *Takeuchi, S., Kanai, T., Suzuki, H., Shimada, K., and Kimura, F*. Subdivision surface fitting with QEM-based mesh simplification and reconstruction of approximated B-spline surfaces. Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on, Page(s): 202 –212, 2000



Original mesh (70,556 triangles) | 8-bit detail encoded (70,508 triangles) | 12-bit detail encoded (70,508 triangles)



Original mesh (96,966 triangles) | 8-bit detail encoded (96,960 triangles) | 12-bit detail encoded (96,960 triangles)



Original mesh (67,170 triangles) | 8-bit detail encoded (67,168 triangles) | 12-bit detail encoded (67,168 triangles)

Figure 6: Comparison of compression results of Bunny, Horse and Igea data sets.

Data #V	Level	#V	% of no. of vertex points (all vertex points require zero displacement)	% of no. of edge points with zero displacement	Total % of no. of points requiring zero displacement
Spock* 16,389	Control mesh	1,039			
	1	4,121	25.21	15.97	41.18
	2	16,417	25.10	62.65	87.75
Igea* 33,587	Control mesh	2,102			
	1	8,398	25.02	8.86	33.88
	2	33,586	25.00	20.47	45.47
Bunny+ 35,280	Control mesh	2,206			
	1	8,818	25.02	13.60	38.62
	2	35,266	25.00	24.34	49.34
Horse* 48,485	Control mesh	3,032			
	1	12,122	25.01	55.02	80.03
	2	48,482	25.00	72.15	97.15

Table 1: Zero displacement at each level (with 8-bit quantization).

Input			**Output**						
Data	#V #T	**Size** (KB)	**Control mesh**			**Details (displacement)**			**Compress** (control mesh + OPTZ)
			#V #T	Size before 3DCP (KB)	Size after 3DCP (KB)	8 bits (KB)	12 bits (KB)	OPTZ (KB)	
Spock *	16,389 32,718	575	1,039 2,044	36.13	3.54	15.87	22.87	6.21	**98.30%**
Igea *	33,587 67,170	1,181	2,102 4,198	73.82	5.85	31.18	46.18	27.15	**97.21%**
Bunny +	35,280 70,556	1,240	2,206 4,408	77.5	6.16	32.5	48.5	26.76	**97.35%**
Horse *	48,485 96,966	1,704	3,032 6,060	106.5	6.21	44.5	67.5	9.25	**99.09%**

Table 2: Quantitative compression results.

Legend:

Data with * courtesy of Cyberware; + courtesy of Stanford Univ. Computer Graphics Lab
#V = Number of vertices
#T = Number of triangles
KB = Kilo-Bytes
3DCP = Lossless compression by *3DCompress.com* software (performed on control- meshes)
OPTZ = Optimized encoding scheme (variable-length encoding)