Subdivision-Based Surface Representations

Denis Zorin* Caltech
Michael Holst† Caltech
Peter Schröder‡ Caltech

Abstract

Subdivision can be thought of as a natural extension of existing patch-based surface representations. In this view, subdivision algorithms effectively bridge the gap between patch-based and polygonal mesh representations. Using subdivision also naturally leads to useful multiresolution representations of surfaces. Such representations can be used for a variety of applications, most importantly, interactive editing and animation. We found that such representations are equally applicable for construction of models from scratch as well as manipulation of scanned high-resolution geometry data.

In our ongoing work we are addressing a few important research problems that have to be solved to make such applications practical. Currently our research is focused on two major issues: the design of efficient algorithms for manipulation of subdivision-based multiresolution surfaces and development of algorithms for fitting of meshes with subdivision connectivity to unorganized surface samples.

In this paper we discuss our approach to these problems, compare them to the related work, and demonstrate some results.

1 High Resolution Meshes

Geometric modeling in special effects production and animation is characterized by the need to create and manipulate complex geometric models of arbitrary topology. These typically contain detail at many scales (cf. Fig. 1). While they are at times constructed from scratch, they are often generated through the use of range sensing devices such as laser scanners. Particularly in the latter case they are often presented as high resolution meshes with hundreds of thousands or millions of polygons.

Manipulating such meshes, especially in interactive modeling or animation environments is a challenge. While it is possible to generate lower resolution approximations through mesh optimization (e.g., [13, 11]), we are interested in representations that can be used for interactive manipulation of data at different levels of resolution; optimization techniques typically are too slow to generate multiple levels of resolution on the fly.

To effectively edit or animate a given high resolution mesh hierarchical representations are necessary: we want to affect large scale smooth changes to the surface shape as easily as minute edits at the individual vertex level. While it is possible to build smoother approximations to the original mesh using patches [12], high frequency detail will be lost. This could be partially recovered through the use of displacement maps [15], but the management of arbitrary topology settings is difficult in this approach.

*dzorin@gg.caltech.edu
†holst@ama.caltech.edu
‡ps@cs.caltech.edu

Figure 1: Before the Armadillo started working out he was flabby, complete with a double chin. Now he exercises regularly. The original is on the right (courtesy Venkat Krischnamurthy, Stanford University). The edited version on the left illustrates large scale edits, such as his belly, and smaller scale edits such as the belly button and his double chin; all edits were performed at about 5 frames per second on an Indigo R10000 Solid Impact.

We propose to use a single underlying representation, hierarchically structured triangular meshes. What is needed is a way to provide coarse level smooth editing semantics as provided in patch based settings. Subdivision provides the connection between fine level polyhedral meshes and patches and is the basis for a system that we are building. Aside from giving us the desired multiresolution control over the surface shape, subdivision algorithms are very simple, and can be implemented using highly adaptive algorithms and datastructures to build a scalable editing system which provides interactive update rates for complex meshes using only moderate hardware resources [22].

2 Hierarchical Representation and Manipulation

The idea of hierarchical editing and adaptive refinement was first pioneered in the ab initio design setting by Forsey and Bartels [8]. They described H-splines, which model finer level detail on top of coarser structures through the use of offset spline patches. To achieve the correct semantics they observed the importance of describing high resolution detail as offsets with respect to a coordinate frame induced by the lower resolution approximation. Lacking an *analysis*, i.e., coarsification, procedure their approach is very dependent on the editing history of a given object and, in any case, is not suitable if the inital geometry is given as a fine resolution mesh of arbitrary topology (although, see [7] for an attempt at analysis). The arbitrary topology challenge in the ab initio design scenario was addressed by Kurihara [16] who used

Doo-Sabin subdivision in a hierarchical editor. However, without adaptive algorithms this approach cannot hope to deal with the exponential space and time requirements of uniform subdivision.

Patches, or their arbitrary topology generalizations, provide a powerful abstraction when building coarse grain, smooth models. In models with fine detail patch representation can quickly overwhelm the user and even high-end hardware resources. At that point a fine polyhedral mesh becomes a more suitable representation (cf. Fig. 2).

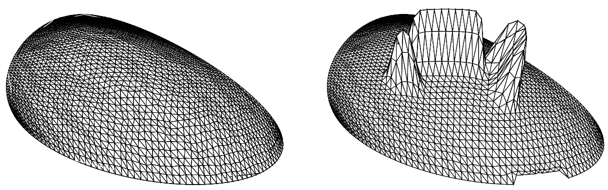

Figure 2: What used to be a patch is best treated as a mesh when adding fine detail.

High resolution polyhedral meshes represent the other end of the spectrum which starts with patches. Such meshes have many advantages, especially if the original model arose from dense measurements such as those produced by a laser range finder. These meshes can resolve fine detail, accommodate arbitrary topology, and can be efficiently rendered on standard graphics hardware. Unfortunately the fine detail also gives rise to so many polygons that even fast polygon rendering hardware cannot display complicated models at interactive rates. Furthermore, it is not immediately clear how to provide coarse level smooth editing semantics.

Subdivision (cf. Fig. 3), originally conceived of as a way to generalize spline based knot insertion [2, 4, 17, 5, 21, 14] to the irregular topology setting, provides the connection we need between fine polyhedral meshes and patch-like, hierarchical editing semantics.

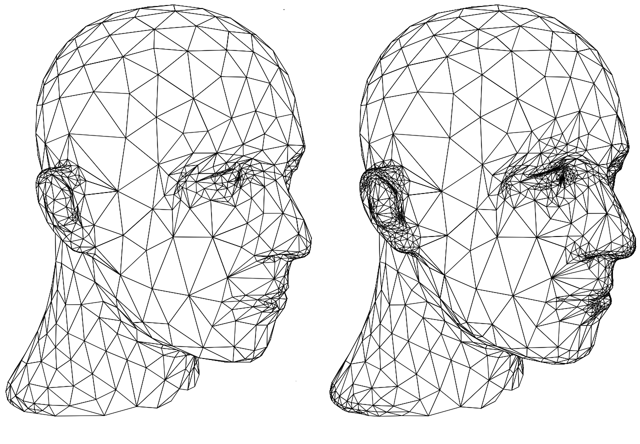

Figure 3: Subdivision describes a smooth surface as the limit of a sequence of refined polyhedra. The meshes show two levels of an adaptive Loop surface generated by our system (Dataset, courtesy Hugues Hoppe, University of Washington)

This connection between subdivision and hierarchical representations of a priori given meshes was first explored in pioneering work by Lounsbery et al. [19]. They built wavelet like multiresolution representations of subdivision connectivity meshes. The wavelet setting provided the needed *analysis* algorithm, which was missing from earlier hierarchical editing work. While their original work only used piecewise linear subdivision, smoother generalizations are possible [20, 21]. Unfortunately the resulting wavelet representations are not suitable for direct editing manipulations, since wavelet coefficients, even when parameterized with respect to local frames, do not behave in intuitive ways [9].

Using instead a Laplacian pyramid [1] type approach overcomes these difficulties. It provides the editing semantics of Forsey and Bartels while using only a mesh, not patches. The fact that the initial mesh must be provided with subdivision connectivity can be addressed through remeshing. Eck et al. [6] have described a possible approach to remeshing arbitrary finest level input meshes fully automatically. A method that relies on a user's expertise was developed by Krishnamurthy and Levoy [15].

With this component of the system in place the generation of meshes with subdivision connectivity is now the major next step towards a complete infrastructure. Instead of first producing a mesh from unstructured samples through a techniques such as marching cubes and then remeshing this mesh, we propose to generate subdivision connectivity meshes directly from the original unstructured point samples generated by some measurement device.

3 Hierarchical Meshes from Unstructured Point Samples

The geometry data generated by scanners typically has the form of unorganized surface samples. A number of approaches were developed for converting such data into a mesh. Curless and Levoy [3], for example, describe a method which collates the original measurements into a signed probability volume. The most common algorithms to turn such a representation into a mesh are based on the *marching cubes* [18] approach.

Algorithms of the type described in Section 2 require meshes with special structure, usually called *meshes with subdivision connectivity*. This means that the mesh can be thought of as a relatively small number of coarsest-level triangles or quadrilaterals, each subdivided uniformly a certain number of times. The importance of obtaining meshes with this structure was recognized in the work of Lounsbery et al. [19].

Eck et al. [6] have described a possible approach to remeshing arbitrary finest level input meshes fully automatically. A method that relies on a user's expertise was developed by Krishnamurthy and Levoy [15]. Several important lessons can be derived from this previous work:

- **Remeshing** Both remeshing methods use an initial mesh obtained from the unorganized samples as a starting point for the algorithm. Aside from additional computational overhead, construction of such a mesh inherently involves arbitrary decisions about connectivity of the mesh, which almost inevitably affect the final results. As pointed out in [15], there is nothing sacrosanct about the original mesh as typically the distance between vertices is close to the resolution of the sensor. It would be desirable to utilize the information in the data directly, without introducing the artifacts of the intermediate meshing step. Furthermore, the marching cubes algorithm is known to produce meshes which are not topological 2-manifolds, thus requiring a special preprocessing step to fix problems with the mesh.

- **User Intervention** The method developed by Eck et al. [6] is fully automatic, while the method of Krishnamurthy and Levoy [15] requires user-specified coarsest-level structure. Both approaches have their advantages and shortcomings. The first approach requires less effort on behalf of the user; however the resulting separation into coarsest-level domains is rather arbitrary, leading to inadequate coarse approximations, introducing artifacts in unpredictable locations of the objects, and unnecessaryly high number of coarsest-level domains. The second approach avoids these problems to the extent the user is able to chose a good partition into the coarsest-level domains. This task can be rather tedious.
- **Patches vs. Subdivision** The final representation used in [6] is the one proposed by Lounsbery and is closely related to the subdivision-based representations that we propose. In [15] a two-level representation consisting of a set of spline patches and displacement maps, is used. The latter representation, while being more suitable for existing modeling and animation systems, has only two levels of resolution and requires different manipulation paradigms for edits performed at the different levels. Manipulation of the geometry near the patch boundaries is difficult and has to be handled as a special case. Adaptive methods are difficult to implement hence it is difficult to use a representation of this type for interactive manipulation, unless the number of knots for the spline patches is relatively small and the displacement map is used as a texture.

The approach to mesh generation from unorganized samples we are currently pursuing attempts to alleviate some of the problems discussed above. We propose the following sequence of steps for generating meshes with hierarchical structure from dense unorganized samples:

- **Limited User Input** The user is required to specify a number of features (boundary curves and vertices) on the object that have to be incorporated into the coarsest-level structure. This set of features does not have to form a complete partition of the mesh. In fact, tagging important coarsest level features is performed before a mesh is constructed.
- **Volume Representation** In a first step the original measurements are used to generate a sampled, signed probability volume as proposed by Curless and Levoy [3].
- **Automatic Generation of the Coarsest Level Topology** A full coarsest-level structure of the object is constructed, using a volume thinning algorithm to derive the skeleton of the object [10]. Using this skeleton a triangulated polyhedron is constructed which additionally incorporates the features specified by the user on the previous stage.
- **Multigrid Fitting** Once the topological structure of the coarsest level is defined, it can be regarded as an abstract domain for the surface. Then the problem of finding a mesh with subdivision connectivity approximating the original set of measurements, can be reformulated as a variational problem. The features specified in the first stage give rise to a set of boundary conditions for the problem. For a suitable functional measuring the quality of the mesh and the precision of the fit, the variational problem leads to a system of non-linear PDEs. A multigrid solution of this system leads directly to a mesh with subdivision connectivity approximating the given set of samples.

Reduction to a variational problem. In this discussion, we will assume that we are constructing a triangular mesh; the approach can be applied to quadrilateral meshes as well.

Once the structure of the coarsest level is defined, we may construct a simple polyhedron topologically equivalent to the mesh that we want to find. The mesh can be thought of as a deformation of a uniformly subdivided initial polyhedron, which need not be embedded in the three-dimensional space and can be regarded as simply a collection of triangles glued together with a suitable metric.

Our problem can be abstracted as a problem of finding a continuous function defined on the initial polyhedron with values in the three-dimensional space, which approximates the given set of samples. Then the mesh that we want to find is a piecewise linear approximation to this function. Powerful multigrid methods can be applied to find an approximate solution of problems of this type.

To specify the variational problem, we need to define a functional that has to be minimized. Our functional consists of two main parts. The first part is simply a measure of a distance function defined on the volume near the target set of samples. This part measures the distance between the solution and the set of samples. In addition to this measure, we need to control the local distortion of the surface. While different strategies are possible, the simplest one would be to produce a mesh with all triangles of approximately equal size and aspect ratio close to one. Assuming that the triangles in the initial polyhedron were chosen to satisfy this requirement, a measure of local distortion can be easily formulated.

Once the functional for the variational problem is obtained, the variational problem can be reduced to a system of PDEs with boundary conditions derived from the features specified by the user on the first stage. A multigrid solution of this system produces a mesh that approximates the target sample set and minimizes the distortion of the mesh in the sense described above.

The main advantages of the method that we propose include

- **Direct Mesh Construction** There is no intermediate mesh construction stage, and, therefore, no artifacts associated with it.
- **Better Control over Topology** Our approach can easily accommodate surfaces that were not sampled adequately in some areas. As the topology of the final mesh is the same as the topology of the initial polyhedron, the method can handle situations when parts of the surface are not adequately sampled; the correct structure of the mesh will be automatically maintained.
- **Flexibility with Respect to User Interaction** While user intervention is required, it is limited to specifying only the most significant features; the user does not have to worry about maintaining correct topological structure, covering the whole surface etc.

4 Results

Using representation of the type described in Section 2 we have developed a set of efficient algorithms for interactive mesh editing. We have implemented a mesh editor based on these algorithms on Pentium Pro and SGI platforms (200 MHz Pentium Pro with Intergraph Intense 3D graphics and Indigo R10000 with Solid Impact graphics). Our system allows interactive editing of meshes of substantial size (hundreds of thousands of triangles). Adaptivity of the underlying algorithms proved to be a crucial feature: meshes of this size could not even be displayed at interactive rates in their entirety with the hardware we were using. Proper choice of resolutions and thresholds for synthesis and rendering allowed us to increase the frame rate to approximately 5 frames per second while retaining a substantial amount of detail in the mesh. Our system handles ab initio design and editing of scanned meshes equally well within a single framework. In addition to the benefits mentioned in Section 2 this feature makes it potentially possible to use scanned meshes as building blocks for new models (such a capability would require a more sophisticated user interface, which would be straightforward to add to our system). An example of an edit produced using our system is shown in Fig. 1. A detailed description of the algorithms and the structure of the system can be found in [22].

We are currently working on a system for mesh generation based on the principles described in Section 3.

5 Conclusions

Using subdivision as a fundamental approach we are able to bridge the gap between patch based modeling and fine polyhedral meshes with high frequency detail. Both in the ab initio design and, even more so, in the scanned geometry setting, finest level meshes are the most versatile. Using highly adaptive algorithms we have built a hierarchical mesh editor which delivers interactive performance across a wide range of hardware resources from PCs to high end workstations. Combined with a tool to build high resolution meshes with subdivision connectivity directly from measurement data a complete system results.

Since the original source of the input data is not fixed by any of the algorithms we use, many applications are possible. For example, the input to the system could come directly from a volumetric imaging technique such as MRI or CT, opening up many applications in biological and bio-medical computing.

Using a subdivision connectivity mesh as the base representation also facilitates many scientific computing tasks which are accelerated by multigrid algorithms. Examples include fast solves for elliptic PDEs over complex surfaces and boundary integral methods using wavelets. Furthermore, the meshes are of a suitable form for compression applications such as those based on zero-tree coders. We intend to pursue these avenues, among others, in the future.

Acknowledgments

We would like to thank Venkat Krishnamurthy for allowing us to use the Armadillo dataset. Andrei Khodakovsky and Gary Wu helped beyond the call of duty to bring the system up. Wim Sweldens helped refine many of the underlying ideas.

The research was supported in part through grants from the Intel Corporation, Microsoft, the Charles Lee Powell Foundation, the Sloan Foundation, and an NSF CAREER award (ASC-9624957) to the third author. Other support was provided by the NSF STC for Computer Graphics and Scientific Visualization.

References

[1] Burt, P. J., and Adelson, E. H. Laplacian Pyramid as a Compact Image Code. *IEEE Trans. Commun. 31*, 4 (1983), 532–540.

[2] Catmull, E., and Clark, J. Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes. *Computer Aided Design 10*, 6 (1978), 350–355.

[3] Curless, B., and Levoy, M. A Volumetric Method for Building Complex Models from Range Images. In *SIGGRAPH 96 Conference Proceedings*, H. Rushmeier, Ed., Annual Conference Series, 303–312, August 1996.

[4] Doo, D., and Sabin, M. Analysis of the Behaviour of Recursive Division Surfaces near Extraordinary Points. *Computer Aided Design 10*, 6 (1978), 356–360.

[5] Dyn, N., Levin, D., and Gregory, J. A. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Trans. Gr. 9*, 2 (April 1990), 160–169.

[6] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. Multiresolution Analysis of Arbitrary Meshes. In *Computer Graphics Proceedings*, Annual Conference Series, 173–182, 1995.

[7] Forsey, D., and Wong, D. Multiresolution Surface Reconstruction for Hierarchical B-splines. Tech. rep., University of British Columbia, 1995.

[8] Forsey, D. R., and Bartels, R. H. Hierarchical B-Spline Refinement. *Computer Graphics (SIGGRAPH '88 Proceedings)*, Vol. 22, No. 4, pp. 205–212, August 1988.

[9] Gortler, S. J., and Cohen, M. F. Hierarchical and Variational Geometric Modeling with Wavelets. In *Proceedings Symposium on Interactive 3D Graphics*, May 1995.

[10] Haralick, R. M., and Shapiro, L. *Computer and Robot Vision*. Addison Wesley, 1993.

[11] Hoppe, H. Progressive Meshes. In *SIGGRAPH 96 Conference Proceedings*, H. Rushmeier, Ed., Annual Conference Series, 99–108, August 1996.

[12] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., and Stuetzle, W. Piecewise Smooth Surface Reconsruction. In *Computer Graphics Proceedings*, Annual Conference Series, 295–302, 1994.

[13] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. Mesh Optimization. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, J. T. Kajiya, Ed., vol. 27, 19–26, August 1993.

[14] Kobbelt, L. Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. In *Proceedings of Eurographics 96*, Computer Graphics Forum, 409–420, 1996.

[15] Krishnamurthy, V., and Levoy, M. Fitting Smooth Surfaces to Dense Polygon Meshes. In *SIGGRAPH 96 Conference Proceedings*, H. Rushmeier, Ed., Annual Conference Series, 313–324, August 1996.

[16] Kurihara, T. Interactive Surface Design Using Recursive Subdivision. In *Proceedings of Communicating with Virtual Worlds*. Springer Verlag, June 1993.

[17] Loop, C. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[18] Lorensen, W. E., and Cline, H. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, M. C. Stone, Ed., vol. 21, 163–169, July 1987.

[19] Lounsbery, M., DeRose, T. D., and Warren, J. Multiresolution Surfaces of Arbitrary Topological Type. Department of Computer Science and Engineering 93-10-05, University of Washington, October 1993. Updated version available as 93-10-05b, January, 1994.

[20] Schröder, P., and Sweldens, W. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics Proceedings, (SIGGRAPH 95)* (1995), 161–172.

[21] Zorin, D., Schröder, P., and Sweldens, W. Interpolating Subdivision for Meshes with Arbitrary Topology. *Computer Graphics Proceedings (SIGGRAPH 96)* (1996), 189–192.

[22] Zorin, D., Sweldens, W., and Schröder, P. Interactive Multiresolution Mesh Editing. Tech. Rep. CS-TR-97-06, California Institute of Technology, 1997. ftp://ftp.cs.caltech.edu/tr/cs-tr-97-06.ps.Z.