Eurographics/ IEEE-VGTC Symposium on Visualization (2007)
Ken Museth, Torsten Möller, and Anders Ynnerman (Editors)

# Subdivision Volume Splatting

K. T. McDonnell[1], Neophytos Neophytou[2], Klaus Mueller[2] and Hong Qin[2]

[1]Department of Mathematics and Computer Science, Dowling College, NY, USA
[2]Center for Visual Computing, Stony Brook University, Stony Brook, NY, USA

**Abstract**

*Volumetric Subdivision (VS) is a powerful paradigm that enables volumetric sculpting and realistic volume deformations that give rise to the concept of "virtual clay". In VS, volumes are commonly represented as a space-filling set of deformed polyhedra, which can be further decomposed into a mesh of tetrahedra for rendering. Images can then be generated via tetrahedral projection or raycasting. A current shortcoming in VS-based operations is the need for a very high level of subdivision to represent fine detail in the mesh and to obtain a high-fidelity visualization. However, we have discovered that the subdivision process itself can be closely simulated with radial basis functions (RBFs), making it possible to replace the finer subdivision levels by a coarser aggregation of RBF kernels. This reduction to a simplified assembly of RBFs subsequently enables interactive rendering of volumetric subdivision shapes within a GPU-based volume splatting framework.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computing Methodologies]: Computer GraphicsPicture/Image Generation;I.3.7 [Computing Methodologies]: Computer GraphicsThree-Dimensional Graphics and Realism ;I.4.10 [Computing Methodologies]: Image ProcessingVolumetric Image Representation;

## 1. Introduction

In this paper we present a rendering framework that employs radial basis functions (RBFs) for the interactive visualization of subdivision volumes. Subdivision volumes comprise a class of mathematical models for representing volumetric objects of arbitrary geometries, topologies and material distributions. They generally have provable continuity properties and are extremely powerful for data and shape representation. For example, they can represent 4D time-varying scalar fields defined over regular and non-regular subdivision meshes and are able to do so in the context of a multiresolution framework [LPD*02]. Dynamic or physically-based subdivision methodologies can be employed in a wide variety of applications involving heterogeneous material distributions, including volumetric sculpting and deformation [MCQ05], mechanical simulation and animation of solid bodies [CGC*02], and fluid flow [WW99]. Many of the recent advances in the field have involved the development of new subdivision schemes [BWX02], non-manifold subdivision volumes [CQ06], and wavelet-based hierarchical subdivision volumes [LGP*04].

In subdivision volumes, polyhedral cells are recursively subdivided according to certain subdivision rules that oftentimes would reproduce a volumetric spline after an infinite number of subdivisions. Thus, these subdivision schemes provide a 3D form of interpolation with a high-quality kernel. Ray tracing cannot generally provide satisfactory solutions for subdivision volume rendering, since the irregular spatial distribution of the voxels interferes with computations fundamental to ray tracing, such as trilinear interpolation of voxel densities. Cell projection [ST90] of the polyhedral cell decomposition, on the other hand, would require subdivision of each deformed cell into many tetrahedra. Such a method would require a great amount of memory to store connectivity information and time, and it would necessarily introduce errors into the volume.

Given the lack of an effective volume visualization technique for rendering subdivision volumes, we have developed an improved visualization approach founded upon RBF-based splatting. In this framework, the computationally expensive subdivision process is replaced with the efficient evaluation of a set of RBFs that closely mimic the behavior of the subdivision algorithm. Furthermore, our technique is not memory intensive since it performs adaptive refine-

ment only in undersampled portions of the volume. A main contribution of our paper is that it shows that the RBF function closely simulates the subdivision rules and therefore can replace recursive subdivision at a given level of detail. This has decisive implications both on storage and on rendering complexity. Finally, our framework is general enough that it can be adapted easily to the visualization of other procedural subdivision volume schemes employing different subdivision weights and space decompositions.

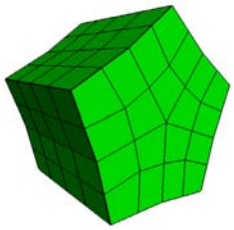

**Figure 1:** *Irregular topological configuration that can be easily accommodated by subdivision volumes but not octrees.*

## 2. Related Work

### 2.1. Subdivision Volume Schemes

Given an initial control mesh consisting of vertices, edges, polygonal faces and polyhedral cells, a volumetric subdivision algorithm describes a recursive procedure for refining the 3D space enclosed by the mesh. For example, a hexahedron-based algorithm typically subdivides each hexahedron into eight smaller hexahedra. Such algorithms essentially produce a polyhedral cell decomposition in which most cells are tetrahedra or hexahedra, depending on the particular algorithm being used. The subdivision rules usually involve taking weighted sums of vertices to generate a new set of vertices. These vertices are then connected to each other according to a pre-determined procedure to generate a refined cell decomposition. In subdivision volumes, these cell decompositions are much more complex than those of subdivision surfaces [CC78].

In this paper we employ the MCQ scheme [MCQ04] for recursive refinement of hexahedral meshes for proof-of-concept experiments. However, as stated in the introduction, our visualization framework is general enough to accommodate any procedural subdivision scheme after its basis function has been accurately fitted to an ellipsoidal reconstruction kernel. The interpolatory MCQ scheme reproduces cubic polynomials for regular topological conditions (i.e., when each vertex is adjacent to exactly six others), and is $C^1$ continuous for irregular topologies. In the regular case, each hexahedron is subdivided into eight smaller hexahedra by computing a set of 27 new vertices: one new point for each existing cell, face, edge and vertex. These 27 new points are then connected to form 8 new hexahedra. These rules are also used to generate the densities in the volumes shown in this paper. Details about the rules for subdivision both hexahedral and non-hexahedral cells can be found in [MCQ04]. An example of an irregular topological setting that can be accommodated by subdivision volumes (but not octrees, for instance) is depicted in Figure 1.

### 2.2. Volume Visualization of Irregular Data

In recent years there has been great interest in developing algorithms for the efficient rendering of tetrahedral meshes and point clouds, with a number of researchers using GPUs to accelerate this process. The majority of these approaches has been based on the projected tetrahedra (PT) algorithm [ST90]. Defining papers in this respect include [RKE00, WME04]. The necessary depth-sorting of the cells and cell faces produces considerable overhead in the PT algorithm. Callahan et al. [CICS05] recently proposed a scheme that partially accelerates this sorting on the GPU by combining a CPU-resident coarse-scale radix-sort with a GPU-resident fine-scale k-buffer. Nevertheless, larger datasets still cannot be rendered at interactive rates, and a number of different strategies have been recently devised to cope with this problem. In another effort, Callahan et al. [CBPS06] use a progressive refinement scheme that first renders the volume boundary for interactive navigation and, once the user pauses, renders the interior in depth order until the full-quality volume image is generated. A year earlier, the same authors introduced a scheme that achieves interactive rendering by reducing the number of cells until the desired frame rate is achieved [CCSS05]. Finally, Cignoni et al. [CFM*04] tackle the problem with a multi-resolution LOD approach that uses an edge-collapse/vertex-split mechanism.

Besides the PT algorithm there have also been approaches that employ RBF-based rendering to project the cells to the screen. This results in a more continuous data representation at the cell boundaries, whereas the PT scheme generates a piece-wise linear result. Using this paradigm, Mao et al. [Mao96], Jang et al. [JRSF02], and Hong et al. [HNMK06] have devised representations that fill or fit each cell with one or more RBFs or EBFs (ellipsoidal basis functions), which are then rendered via generic splatting [Wes90].

While these RBF methods still adhere to the cell decomposition of the data, another way to look at irregular data is to ignore the grid structure completely and consider the grid points as a collection of atomic scattered data points. This eliminates the explicit point connectivity expressed by the unstructured grid's cell links, which may be meaningful in the context of the underlying computational process that generated the grid values. On the other hand, there are also irregular datasets that are inherently connection-less point clouds. While Meredith and Ma [MM01] have employed the grid structure, if available, to devise their multi-resolution RBF-rendering framework, others have used a

signal-processing-inspired method based on Gabor filters to create such a hierarchy [WM03]. Both of these can be considered local grid reduction methods, and they are reasonably fast in terms of runtime speed. In some sense they fall into the same category as the local cell filling and fitting methods mentioned above. Other reduction methods seek to optimize the number of scattered data points in a more global sense [WBS*05] by using an iterative optimization method, such as the Levenberg-Marquardt algorithm. Here, the fit is obtained by minimizing the error of the volume's RBF-encoding at the existing scattered data points. Our approach can be considered a local fitting method, but at the same time it has the error characteristics of the global optimization approaches.

To visualize the field of scattered RBFs of mixed radii, the best images (in terms of quality) are achieved when compositing a set of image-aligned slices, interpolated from the volume, in a post-shaded rendering pipeline. Jang et al. [JWH*04] have recently described such a GPU-accelerated rendering engine, but they use a GPU fragment program to evaluate the Gaussian kernel function of all kernels that overlap at a given point. It turns out that this pixel-centric rendering paradigm using a fragment shader-based kernel evaluation of all RBFs targeting a given pixel tends to be significantly less efficient than the inverse method – an image-aligned RBF-centric rasterization pipeline. A GPU-accelerated implementation of such a strategy was recently described by Neophytou et al. [NMM*06], and this method can also deal with scattered elliptical radial basis functions (EBFs). We use this renderer for the work reported in this paper.

## 3. Specifics on RBF Fitting and its Implication on Rendering

As mentioned above, our RBF-based approach can be considered a local fitting method, yielding the inherent efficiency advantages, but at the same time it has the favorable error characteristics of the global optimization approaches. We achieve this by exploiting the unique set of rules governing the subdivision process. By fitting RBFs, we avoid the need to divide the polyhedral cells into tetrahedra to render the subdivision result. Instead, the RBFs allow us to capture the recursive application of the subdivision rules with high accuracy. Having identified such an RBF kernel then enables us to bypass the calculation of these refinements and the efforts associated with the rendering of the resulting primitives. Instead we can render the much coarser field of RBFs at great efficiency on the GPU.

Our finding is important, as in most subdivision schemes, the number of polyhedra grows exponentially with subdivision level (e.g., approximately $8^n$ in the case of the MCQ algorithm, where $n$ is the subdivision level). (A notable exception is Pascucci's scheme [Pas02].) Thus, for the examples shown in this paper, a single RBF is rendered in place of 8, 64, 512, etc. hexahedral cells, depending on the subdivision level. If we used the PT algorithm for rendering, it would then be necessary to tessellate each of those new hexahedra into 5 or 6 new tetrahedra. This would produce even more cells. In fact, many of the subdivision schemes generate hexahedra, octahedra, decahedra and other multi-faceted polyhedra, which need tetrahedral tessellation. Although it is certainly possible to subdivide complex polyhedra into a collection of smaller tetrahedra, the required linear interpolations will also necessarily introduce error, and to minimize the interpolation errors, it would be necessary to subdivide the volume at least several times. This would again dramatically increase the tetrahedron count as the polyhedron count.

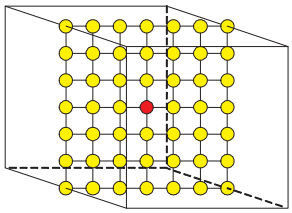

**Figure 2:** *Center slice of the $7 \times 7 \times 7$ mesh of control vertices used to approximate the subdivision kernel. All $7^3$ vertices except the center one are assigned a density of 0.0, while the center vertex, drawn in red, is assigned a weight of 1.0. This mesh, after several subdivisions, provides a discrete representation of the subdivision basis function.*

### 3.1. Approximating Subdivision with RBF Kernels

A key observation for our method is that if we study the behavior of a subdivision algorithm (surface or volume) after an infinite number of subdivisions of an initial control mesh, we often find that in the limit of the subdivision process a spline is generated. Stam [Sta98] showed that for irregular subdivision *surface* control meshes it is often possible to find a closed-form expression for the basis functions. On the other hand, Chang et al. [CMQ02] have shown that, in general, it is impossible to find a closed-form expression for the basis functions of a *volumetric* subdivision scheme when the control mesh is irregular. In a nutshell, the spectral analysis tools used to analyze 2D basis functions do not generalize directly to 3D. Hence, we must resort to approximation techniques, as we shall describe now.

In our framework the basis functions for virtually any procedural subdivision scheme can be approximated in a straightforward manner and be represented in discretized form. In the case of the interpolatory subdivision solid algorithm used as an example in this paper, each vertex in the control mesh affects nearly all of the cells located within the three-neighborhood of the vertex. Hence, to compute the basis function for a vertex, we assign a value of 1 to the center vertex in a $7 \times 7 \times 7$ mesh of vertices and 0 to the other

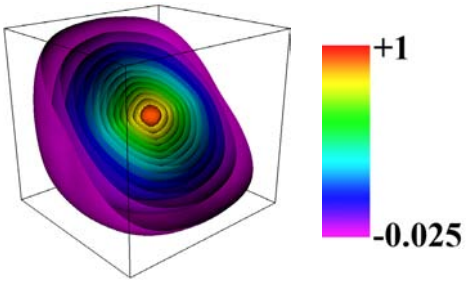


**Figure 3:** *Isosurface rendering showing the central region of the MCQ subdivision algorithm's basis function.*

vertices. A portion of this mesh is depicted in Figure 2. By carrying this process to the limit, we could theoretically obtain the scheme's volumetric basis function for hexahedral meshes of regular topology. In practice, however, we can terminate the subdivision process at level 5, thereby generating a $193^3$ volume. The voxels in this volume are point samples of the 3D continuous basis function, shown in Figure 3.

In our own experiments, we fit the discretized basis function to three radially symmetric, volumetric filters to determine which most closely matched the subdivision scheme's basis function. These filters included a Gaussian distribution:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

a cubic BC-spline filter:

$$h_{BC}(x) = \begin{cases} (2-\frac{3}{2}B-C)|x|^3+ & \\ \quad (-3+2B+C)|x|^2+ & \text{if } |x|<r \\ \quad (1-\frac{1}{3}B) & \\ (-\frac{1}{6}B-C)|x|^3+ & \\ \quad (B+5C)|x|^2+ & \text{if } r\le|x|<2r \\ \quad (-2B-8C)|x|+(\frac{4}{3}B+4C) & \\ 0 & \text{otherwise} \end{cases}$$

and a cubic Cardinal spline filter:

$$h_a(x) = \begin{cases} (a+2)|x|^3-(a+3)|x|^2+1 & \text{if } |x|<r \\ a|x|^3-5a|x|^2+8a|x|-4a & \text{if } r\le|x|<2r \\ 0 & \text{otherwise} \end{cases}$$

Although the subdivision algorithm's basis function is not radially symmetric, as can be seen in Figure 3, we found that it could be approximated quite effectively by a radially symmetric cubic spline. We performed least-squares fittings to compute the spline parameters for the two spline filters. For the (normalized) Gaussian distribution, we assigned a mean of 0 and determined numerically via the Bisection Method that a standard deviation of 0.43 provides an accurate fit to the subdivision basis function. Since the control mesh is of size $7\times7\times7$ and the control points are spaced at unit intervals, we assume that all points in the subdivided mesh are in the volumetric space spanned by $[-3,3]\times[-3,3]\times[-3,3]$.

The fitting process was applied for each type of filter for radii of 1, 2 and 3, resulting in a total of seven trials. The resulting spline parameters and associated errors are summarized in Table 1. To compute the mean squared errors of the fits, we computed the average squared difference between the densities of the sample points in the discretized subdivision kernel and the discretized analytic kernel over all $193^3$ sample values. Visualizations of slices of the kernels of extent 1.0 are provided in Figure 4. Undulations in the basis function outside a radius of 1.0 unit the cubic splines result in moderate levels of error in the fitting. As indicated in the table, we found that the Gaussian kernel provided the best fit.

Since the MCQ subdivision scheme is derived from cubic Lagrange polynomials [MCQ04], it comes as no surprise that a cubic spline can provide a good approximation of the true basis function. However, it should be noted that the original subdivision scheme was derived via tensor-product application of Dyn et al.'s original 4-point scheme for subdivision curve generation [DLG87]. This accounts for the square-like distribution of densities outside a radius of 1.0 in the subdivision kernel. As can be seen in the difference maps in Figure 4, it is in these regions that the kernel approximations show the highest error. This is the price that must be paid for using a mathematically simpler and rotationally symmetric kernel.

### 3.2. Rendering

To perform true and accurate volume rendering of a subdivision volume, it would be necessary to subdivide the dataset a sufficient number of times such that at any zoom level there would be no discernable gaps between sample points. Instead of subdividing many times, we use an adaptive form of the RBF-based approximation just derived to refine the dataset only in those regions where we are zooming in to view the volume or in those regions where the original volume is undersampled. The idea is to refine only large hexahedra, whose image-plane projection would take up many pixels. As illustrated in 2D in Figure 5, we employ interpolation to fill the cell with new sample points until the maximum inter-sample distance between any two neighboring samples is approximately 1.5 units. (Here, a "unit" refers to the length of a voxel. In this paper, we assume each volume is bounded approximately by a $200\times200\times200$ cube of voxels.) The sample positions are determined by trilinear interpolation, while the density values themselves are calculated by centering a reconstruction kernel over a sample point and taking a weighted sum of the original data values that lie within the extent of the RBF. (The extent of the kernel is set to be sufficiently large that it includes all of the vertices of the cell that contains the sample point.) The weights are de-

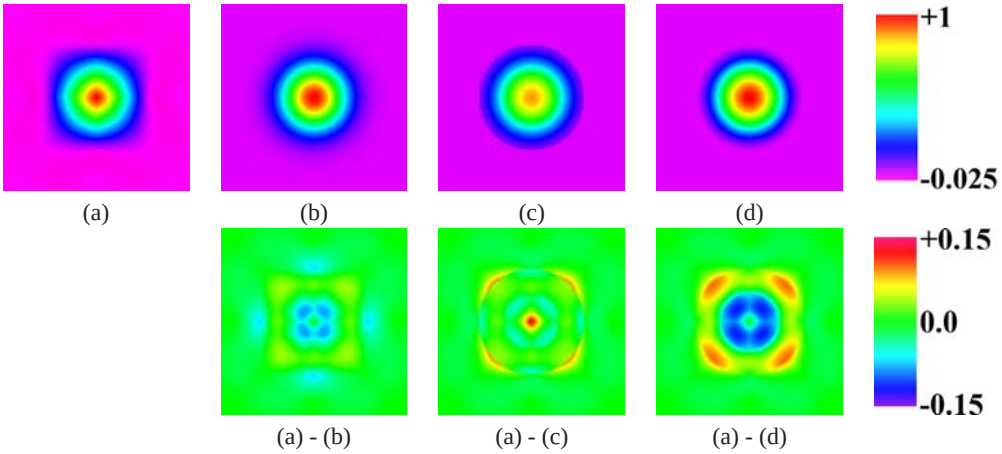


**Figure 4:** *(a) Slice through the center of the subdivision kernel. (b)-(d) Slices through fitted Gaussian, BC-spline and cardinal splines, respectively, for an extent/radius of 1.0. The bottom three images show the differences between a slice through the center of the true kernel and its approximations. The minimum and maximum values of the kernel are approximately -0.022 and 1.0, respectively.*

| Filter | Radius | Parameters | MSE | RMSE | MAE |
|---|---|---|---|---|---|
| **Gaussian** | N/A | $\mu =$ 0.000000<br>$\sigma =$ 0.430000 | 0.000041 | 0.006395 | 0.002079 |
| **BC-spline** | 1.0 | $B =$ 0.382175<br>$C =$ 0.253253 | 0.000054 | 0.007339 | 0.002144 |
| | 2.0 | $B =$ 0.039579<br>$C =$ 0.028746 | 0.000139 | 0.011808 | 0.002892 |
| | 3.0 | $B = -0.062800$<br>$C =$ 0.012591 | 0.011781 | 0.108542 | 0.022184 |
| **Cardinal Spline** | 1.0 | $a = -0.507829$ | 0.000127 | 0.011275 | 0.002895 |
| | 2.0 | $a = -0.098673$ | 0.000156 | 0.012492 | 0.002944 |
| | 3.0 | $a = -0.040698$ | 0.010297 | 0.101472 | 0.028730 |

**Table 1:** *Function parameters and fitting errors for different spline approximations of the basis function for the MCQ interpolatory subdivision solid algorithm. MSE = "Mean Squared Error," RMSE = "Root Mean Squared Error," MAE = "Mean Absolute Error."*

termined in terms of their distances from the center of the RBF, using one of the kernels described in Section 3.1 and the parameters from Table 1.

Splatting an adaptively sampled volume essentially simulates the act of subdividing the volume an infinite number of times. This is the key advantage of using a spline approximation of the subdivision scheme's basis function over explicit subdivision. This is the first time, to our knowledge, that splatting or any other volume visualization algorithm has been used as a vehicle for simulating the recursive refinement of irregular volumetric data.

### 3.3. Benefits of RBF-Driven Subdivision Volume Visualization

**Decreased Memory and Computation Costs.** Subdivision can be circumvented completely at the cost of small error by interpolating densities at desired 3D positions inside cells. By doing so we save memory since the complex cell decomposition does not need to be maintained. Also, we save computation time since it is unnecessary to traverse the cell structure at the current subdivision level or to construct the structure for the next subdivision level.

**User-Specified or System-Specified Sampling Rates.** A global subdivision always introduces a given number of new vertices at locations determined by the particular subdivision scheme. If we use the MCQ scheme as an example, a single application of the subdivision algorithm replaces the 8 vertices of one hexahedron with 27 new vertices. In general, $N$ subdivisions result in the creation of approximately $2^{3N}$ new vertices per hexahedron. Suppose that to satisfy a given error tolerance it is necessary to resample a hexahedron at a final resolution of $5 \times 5 \times 5$. Without our resampling framework,

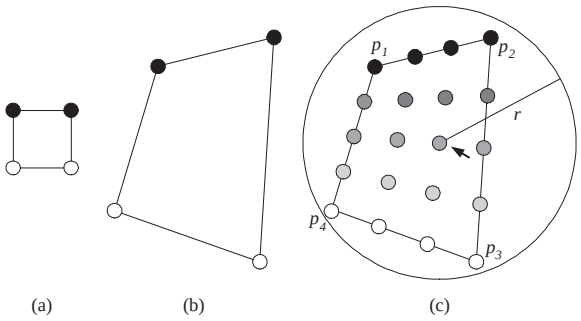


**Figure 5:** *Example of material density reconstruction in 2D. An analogous process is employed in 3D. Standard and deformed cell geometries are shown in (a) and (b), respectively. In (c), a resampled cell is shown, with new sample points drawn in varying shades of gray. The RBF reconstruction kernel of radius r for a particular sample is rendered as an overlaid circle. A weighted sum of the four data points $p_i$ is used to assign material attributes to the sample.*

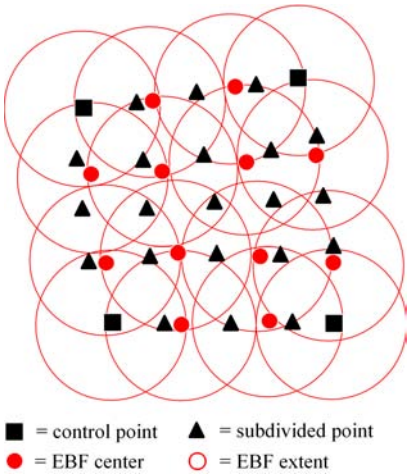


**Figure 6:** *Resampling scheme used to compute RMSE of models. "Subdivided point" refers to a vertex generated by subdivision. Note that each control point has one RBF associated with it.*

it would be necessary to subdivide to level 3, meaning that each hexahedron would contribute $8^3$ vertices. This would result in oversampling and the computation of $(8^3 - 4^3)$ or 448 superfluous samples per hexahedron. (We use $4^3$ in the calculation instead of $5^3$ so as to account for the sharing of samples between adjacent hexahedra.) This process would be repeated over the entire control mesh, wasting considerable time and memory.

**Non-Uniform Sampling Rates.** Explicit, global subdivision of a volume introduces new vertices along all edges and faces. For cells that are long and skinny, it is wasteful to subdivide along those directions corresponding to closely-packed vertices. When projected, the splats for these vertices will overlap significantly. Our resampling framework facilitates adaptive sampling across edges, faces and cells and resamples only where needed. An example of this procedure can be seen in Figure 5(c), where the quadrilateral has been sampled at different rates along edges.

## 4. Results and Error Analysis

We have employed the GPU-accelerated splatting renderer for scattered data, described in close detail in [NMM*06], to test and visualize our RBF-based subdivision volumes. Some examples are given in Figure 7, which shows visualizations of several subdivision volumes that underwent dramatic deformations. Detailed error metrics and performance statistics are given in Table 2. "RMSE" refers to the root mean squared error of the RBF-based interpolation performed over each dataset (both deformed and undeformed). It was computed by subdividing the volume once, taking the squared difference between (i) the density at each new vertex as computed by the subdivision scheme and (ii) the density computed by the RBF resampling scheme described in Section 3.2 and illustrated in Figure 6. We see that the root mean-squared errors are very low and fall in the range $0.003 - 0.011$.

## 5. Conclusions

This paper has presented an efficient and novel framework for the visualization of subdivision volumes. We have demonstrated that splatting in particular is the ideal subdivision volume visualization algorithm since it can be readily adapted to the visualization of the irregular datasets generally produced by subdivision schemes. By employing RBFs to approximate the subdivision process, we greatly reduce the computational cost and memory requirements that would be otherwise required to render these irregular datasets. Our fitting approach, for subdivision volume rendering, replaced the need to render 320 tetrahedral cells with the overhead associated for rendering 125 sliced RBF kernels for every cell, amounting to a ratio of 39% (for 2 levels) and 729 RBFs/2560 tetrahedra or 28% for 3 levels. Possible directions for future work include extension of the framework to encode and visualize time-varying subdivision volumes, application of the visualization framework in interactive volumetric sculpting, and exploitation of the hierarchical nature of subdivision volumes to permit the visualization of larger datasets. We would also like to explore local density-based data reduction techniques, which we have not done so far. As Jang et al. [JWH*04] have demonstrated, the higher-order kernel functions of RBFs allow for aggressive data reduction, more than the piece-wise linear function approximation of cell-based decompositions. We hope to achieve further speedups based on such simplifications. Concomitant with

| Model | RMSE for RBF Fit of Deformed Data | # Data Points | # Points Rendered | Frame Rate |
|---|---|---|---|---|
| Deformed Pawn | 0.0037 | 543,599 | 90,497 | 4.0 |
| Alien | 0.0030 | 487,071 | 31,573 | 4.4 |
| Train #1 | 0.0045 | 1,898,312 | 228,018 | 2.2 |
| Toy Car | 0.0107 | 1,406,847 | 176,802 | 2.7 |
| CSG Fan | 0.0042 | 517,153 | 35,832 | 5.1 |
| Nozzle | N/A | 155,808 | 155,808 | 4.0 |
| Train #2 | 0.0044 | 1,636,349 | 216,476 | 2.5 |

**Table 2:** *Fitting errors and rendering speeds for the models shown in this paper. Densities lie in the range [-1,+1] and were approximated using a cubic BC-spline of radius 2 with $B = 0.039579$ and $C = 0.028746$. One level of subdivision was performed on the original volume. The nozzle dataset was generated via physical simulation, and so no RBF fitting was possible.*

these efforts should be further investigation of spline encoding techniques, including the possibility of using wavelet analysis to reduce fit error even further.

## References


[BWX02] BAJAJ C., WARREN J., XU G.: A subdivision scheme for hexahedral meshes. *The Visual Computer 18* (2002), 343–356.

[CBPS06] CALLAHAN S. P., BAVOIL L., PASCUCCI V., SILVA C. T.: Progressive volume rendering of large unstructured grids. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2006) 12*, 5 (Sept/Oct 2006), 1307–1314.

[CC78] CATMULL E., CLARK J.: Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10* (Sept. 1978), 350–355.

[CCSS05] CALLAHAN S. P., COMBA J. L. D., SHIRLEY P., SILVA C. T.: Interactive rendering of large unstructured grids using dynamic level-of-detail. In *Proceedings of IEEE Visualization 2005* (2005), pp. 199–207.

[CFM*04] CIGNONI P., FLORIANI L. D., MAGILLO P., PUPPO E., SCOPIGNO R.: Selective refinement queries for volume visualization of unstructured tetrahedral meshes. *IEEE Transactions on Visualization and Computer Graphics 10*, 1 (2004), 29–45.

[CGC*02] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIC Z.: A multiresolution framework for dynamic deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 41–47.

[CICS05] CALLAHAN S. P., IKITS M., COMBA J. L., SILVA C. T.: Hardware-assisted visibility sorting for unstructured volume rendering. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 285–295.

[CMQ02] CHANG Y., MCDONNELL K. T., QIN H.: A new solid subdivision scheme based on Box splines. In *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications* (June 2002), pp. 226–233.

[CQ06] CHANG Y.-S., QIN H.: A unified subdivision approach for multi-dimensional non-manifold modeling. *Computer-Aided Design 38*, 7 (2006), 770–785.

[DLG87] DYN N., LEVIN D., GREGORY J.: A four-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design 4*, 4 (1987), 257–268.

[HNMK06] HONG W., NEOPHYTOU N., MUELLER K., KAUFMAN A.: Constructing 3D elliptical gaussians for irregular data. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration* (2006), Möller T., (Ed.), Springer. To appear.

[JRSF02] JANG J., RIBARSKY W., SHAW C. D., FAUST N.: View-dependent multiresolution splatting of non-uniform data. In *Proceedings of the 2002 Symposium on Data Visualisation* (2002), pp. 125–132.

[JWH*04] JANG Y., WEILER M., HOPF M., HUANG J., EBERT D. S., GAITHER K. P., ERTL T.: Interactively visualizing procedurally encoded scalar fields. In *Proceedings of the EG/IEEE TCVG Symposium on Visualization (VisSym '04)* (2004), pp. 35–44.

[LGP*04] LINSEN L., GRAY J., PASCUCCI V., DUCHAINEAU M. A., HAMANN B., JOY K. I.: Hierarchical large-scale volume representation with $\sqrt[3]{2}$ subdivision and trivariate B-spline wavelets. In *Geometric Modeling for Scientific Visualization*, Brunnett G., Hamann B., Mueller H., Linsen L., (Eds.). Springer Verlag, 2004.

[LPD*02] LINSEN L., PASCUCCI V., DUCHAINEAU M. A., HAMANN B., JOY K. I.: Hierarchical representation of time-varying volume data with $\sqrt[4]{2}$ subdivision and quadrilinear B-spline wavelets. In *Proceedings of Tenth Pacific Conference on Computer Graphics and Applications* (2002), pp. 346–355.

[Mao96] MAO X.: Splatting of non-rectilinear volumes through stochastic resampling. *IEEE Transactions on Visualization and Computer Graphics 2*, 2 (1996), 156–170.

[MCQ04] MCDONNELL K. T., CHANG Y., QIN H.: In-

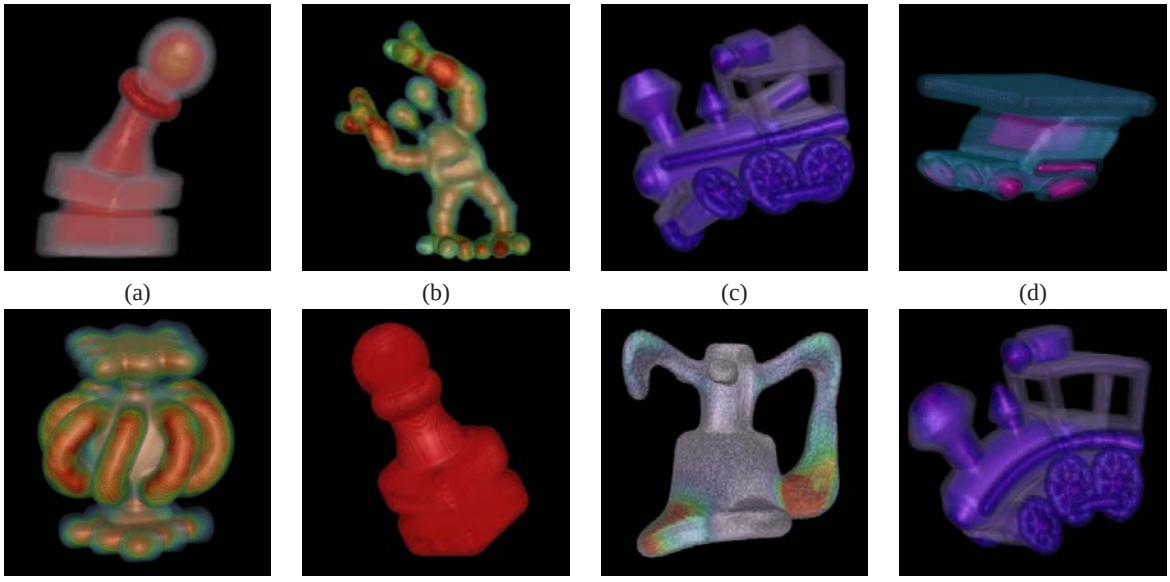


**Figure 7:** *Rendering results: (a) Deformed Pawn, (b) Alien, (c) Train #1, (d) Deformed Toy Car, (e) CSG Fan, (f) Original Chess Pawn, (g) Deformed Nozzle, (h) Train #2*

terpolatory, solid subdivision of unstructured hexahedral meshes. *The Visual Computer 20*, 6 (2004), 418–436.

[MCQ05] MCDONNELL K. T., CHANG Y., QIN H.: *DigitalSculpture*: Interactive implicit surface modeling over 3D unstructured meshes. *Graphical Models 67*, 4 (2005), 347–369.

[MM01] MEREDITH J., MA K.-L.: Multiresolution view-dependent splat based volume rendering of large irregular data. In *Proccedings of the IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics* (2001), pp. 93–99.

[NMM*06] NEOPHYTOU N., MUELLER K., MCDONNELL K., HONG W., GUAN X., QIN H., KAUFMAN A.: GPU-accelerated volume splatting with elliptical RBFs. In *Proceedings of the Joint Eurographics-IEEE TCVG Symposium on Visualization 2006 (EuroVis '06)* (2006), pp. 13–20.

[Pas02] PASCUCCI V.: Slow growing subdivision (SGS) in any dimension: Towards removing the curse of dimensionality. *Computer Graphics Forum (Proceedings of Eurographics 2002) 21*, 3 (Sept. 2002), 451–460.

[RKE00] ROETTGER S., KRAUS M., ERTL T.: Hardware-accelerated volume and isosurface rendering based on cell-projection. In *Proceedings of IEEE Visualization 2000* (2000), pp. 109–116.

[ST90] SHIRLEY P., TUCHMAN A.: A polygonal approximation to direct scalar volume rendering. In *Computer Graphics (San Diego Workshop on Volume Visualization)* (Nov. 1990), pp. 63–70.

[Sta98] STAM J.: Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. In *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH* (July 1998), pp. 395–404.

[WBS*05] WEILER M., BOTCHEN R. P., STEGMEIER S., ERTL T., HUANG J., JANG Y., EBERT D. S., GAITHER K. P.: Hardware-assisted feature analysis of procedurally encoded multifield volumetric data. *Computer Graphics and Applications 25*, 5 (2005), 72–81.

[Wes90] WESTOVER L.: Footprint evaluation for volume rendering. In *Computer Graphics (Proceedings of ACM SIGRRAPH 90)* (Aug. 1990), vol. 24, pp. 367–376.

[WM03] WELSH T., MUELLER K.: A frequency-sensitive point hierarchy for images and volumes. In *Proceedings of IEEE Visualization 2003* (2003), pp. 425–432.

[WME04] WEILER M., MALLON P. N., ERTL M. K. T.: Texture-encoded tetrahedral strips. In *Proceedings of the 2004 IEEE Symposium on Volume Visualization* (2004), pp. 71–78.

[WW99] WEIMER H., WARREN J.: Subdivision schemes for fluid flow. In *SIGGRAPH '99 Computer Graphics Proceedings, Annual Conference Series* (Aug. 1999), pp. 111–120.