Nested Refinement Domains for Tetrahedral and Diamond Hierarchies

Kenneth Weiss*
University of Maryland, College Park

Leila De Floriani†
Università di Genova, Italy

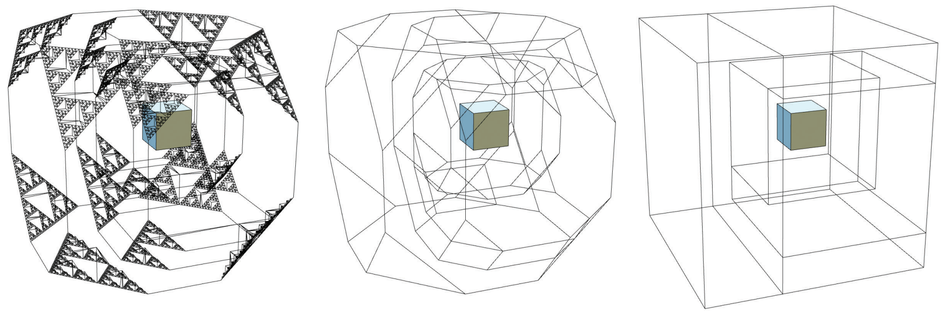


Figure 1: Three *nested refinement domains* for hierarchies of tetrahedra and diamonds. The *descendant domain* (left) is the limit shape of the domain covered by all descendants of a given diamond (colored). Due to the fractal nature of these shapes, we introduce the more conservative *convex descendant domain* (middle) and *bounding box descendant domain* (right) to simplify the computation while still tightly covering the descendant domain. In each case, the refinement domain of one of the diamond's parents, grandparents and great-grandparents is shown.

Abstract

We investigate several families of polyhedra defining *nested refinement domains* for hierarchies generated through longest edge tetrahedral bisection. We define the *descendant domain* of a tetrahedron as the domain covered by all possible descendants generated by *conforming* bisections. Due to the fractal nature of these shapes, we propose two simpler approximations to the descendant domain that are relatively tight with respect to the descendant domain and can be implicitly computed at runtime. We conclude with a brief discussion of the applications of these shapes for interactive view–dependent volume visualization and isosurface extraction.

1 Introduction

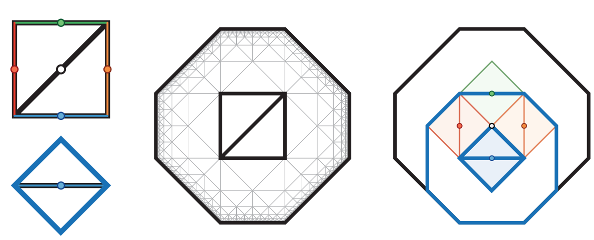
Hierarchical domain decompositions play a fundamental role in analysis and visualization within scientific and mathematical computing. Such decompositions are typically driven by an application–dependent *selection criterion*, and can be generated *bottom-up*, by simplifying a fine mesh, *top-down*, by refining a coarse mesh, or *incrementally*, via simplifications and refinements.

A class of such decompositions that has proven to be particularly effective for visualization applications is based on *nested* hierarchies of tetrahedra generated by *longest edge bisection (LEB)*, in which a single *parent* tetrahedron is replaced by two *child* tetrahedra created during its bisection. When initialized over a cubic domain that has been tetrahedralized through a diagonal, this process generates only three classes of tetrahedra. The nesting property is significant, since it enables simple top-down spatial selection queries. However, tetrahedral bisection, on its own, does not generate *conforming* domain decompositions, i.e. it introduces cracks between neighboring tetrahedra. This is problematic since it can lead to discontinuities in functions defined on the domain.

Conforming bisections along an edge **e** can be achieved by ensuring that **e** is the longest edge of all tetrahedra incident to it, and by concurrently bisecting these tetrahedra. This can be satisfied at runtime by recursively bisecting neighbors that do not share this property or by utilizing a geometric data structure, typically referred to as a *diamond* (see Figures 2a and 3a), defined by the set of tetrahedra sharing a common longest edge. See [7] for a recent survey on simplex and diamond hierarchies and their applications.

Unfortunately, the hierarchical dependency relationships required for conforming bisections no longer defines a nested hierarchy. One approach to reintroduce a nested hierarchy is to explicitly integrate such dependencies into a *saturated* selection criterion, in which a tetrahedron's error must be greater than those of its LEB neighbors as well as their children (see [3] and references therein).

More flexible approaches have been proposed for generating conforming decompositions over two dimensional domains. In particular, each triangle, or diamond, can be associated with a simple geometric primitive, such that the domain covered by this shape is enclosed by the shape(s) associated with its parent(s). Blow [2] introduces an explicit nested hierarchy of spheres for view-dependent rendering of terrain datasets that is independent from the diamond hierarchy. Lindstrom et al. [4] incorporate this spherical hierarchy into the diamond hierarchy. Tanaka [5], Balmelli [1] and Gerstner [3] consider a nested hierarchy based on a node's descendants. In particular, the limit shape of the domain covered by a diamond's descendants is an octagon whose edge lengths are defined by the diamond's triangles (see Figure 2b). This space is the tightest possible nesting domain since every diamond's octagon is entirely covered by those of its children (see Figure 2c). In an attempt to generalize the octagon to 3D, Tanaka et al. [6] propose the *rhombicuboctahedron*, which would not ensure crack-free decompositions.



(a) (b) (c)

Figure 2: 2D diamonds and their octagonal descendant domains.

*kweiss@cs.umd.edu
†deflo@disi.unige.it

2 Nested Refinement Domains

In this Section, we generalize the 2D refinement domains of [5, 1, 3] to 3D by introducing three families of polyhedra defining nested refinement domains. We illustrate the nestedness of these shapes in Figure 1 by showing how the refinement domain of a single diamond lies in those of its ancestors up to three levels higher.

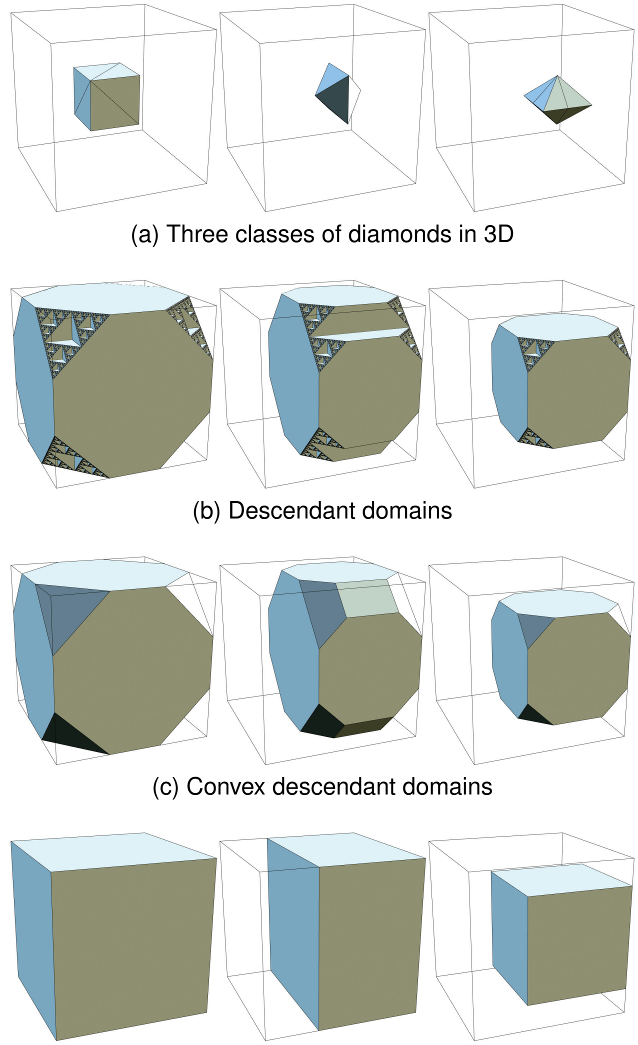
We define the *descendant domain* of a tetrahedron in a hierarchy of tetrahedra as the limit shape of the domain covered by all possible descendants generated by conforming bisections. Since all tetrahedra in a diamond refine concurrently, they share the same descendant domain, and we can equivalently discuss the descendant domain of a diamond. It is evident from the definition that the descendant domain of a diamond can be defined recursively as the union of the descendant domains of its children. As such, these form a nested hierarchy for conforming decompositions.

Within each diamond's descendant domain, the faces that are aligned with the coordinate planes are octagonal (or trapezoidal sections of octagons) and have the same proportions as their two dimensional counterparts (compare Figure 3b to Figure 2b).

Interestingly, the triangular faces on the boundary of the domain have a Sierpinski-like fractal refinement. That is, the refinement is generated through quaternary refinement. However, instead of being removed, the internal triangles are trisected, and the midpoint is offset along its normal by a distance of $\sqrt{6}/6$ of the triangle's edge length. The indentations formed by this process are cube corners aligned with the coordinate axes.

The fractal nature of the above descendant domain makes it difficult to compute with in a top-down manner. As such, we define a second family of nested refinement domains obtained by taking the convex hulls of the above shapes, which we refer to as the *convex descendant domains* (see Figure 3c). Combinatorially, these polyhedra are *truncated cuboids* (with four beveled edges in one case).

We observe that the descendant domain almost defines a hierarchy of nested *cuboids*. We therefore introduce the *bounding box descendant domains*. The dimensions of these cuboids, relative to a unit 0-diamond (i.e. covering a unit cube), are $3 \times 3 \times 3$ (Figure 3d, left), $3 \times 3 \times 2$ (Figure 3d, middle, where the shorter dimension is orthogonal to the bisection edge), and $2 \times 2 \times 2$ (Figure 3d, right).



(a) Three classes of diamonds in 3D

(b) Descendant domains

(c) Convex descendant domains

(d) Bounding box descendant domains

Figure 3: The three classes of diamonds (a) and their corresponding nested refinement domains (b-d) in 3D.

3 Concluding Remarks

In this paper, we introduce three nested refinement domains for tetrahedral and diamond hierarchies. The *descendant domain*, is the 3D analogue of the octagonal-shaped domains of [5, 1, 3]. However, in contrast to the 2D case, these descendant domains have a fractal boundary. We therefore introduce the *convex descendant domain* and *bounding box descendant domain* refinement hierarchies, which are easier to compute with at runtime.

These hierarchies have several potential applications to interactive volume visualization, which we intend to explore in future work. As with the 2D case, they can be easily incorporated into a view–dependent visualization system for frustum culling. Observe that the bounding box domain, which incorporates the other two refinement domains, defines a relatively small inflation factor for each diamond. Thus, only diamonds (or tetrahedra) within a distance of at least three *units* from the frustum boundaries need be checked, where units refer to the scale of the diamond. Furthermore, as pointed out by Blow [2], once a diamond is outside this distance, its descendants no longer need to be checked.

The bounding box refinement domain has similar implications for isosurface queries. The range of isovalues covered by all descendants can be conservatively approximated from a non-nested min/max metric (such as BONO [8]) by considering the isovalue range of a constant number of cubes.

Acknowledgements

This work has been partially supported by the National Science Foundation under grant CCF-0541032.

References

[1] L. Balmelli, S. Ayer, and M. Vetterli. Efficient algorithms for embedded rendering of terrain models. In *Proceeding ICIP*, pages 914–918, 1998.

[2] J. Blow. Terrain rendering at high levels of detail. In *GDC*, 2000.

[3] T. Gerstner. Top-down view-dependent terrain triangulation using the octagon metric. Technical report, University of Bonn, 2003.

[4] P. Lindstrom and V. Pascucci. Terrain simplification simplified. *IEEE Trans. on Visualization and Computer Graphics*, 8(3):239–254, 2002.

[5] H. Tanaka. Accuracy-based sampling and reconstruction with adaptive meshes for parallel hierarchical triangulation. *Computer Vision and Image Understanding*, 61(3):335 – 350, 1995.

[6] H. Tanaka, Y. Takama, and H. Wakabayashi. Accuracy-based sampling and reconstruction with adaptive grid for parallel hierarchical tetrahedrization. In *Proceedings Volume Graphics*, pages 79–86, 2003.

[7] K. Weiss and L. De Floriani. Simplex and diamond hierarchies: Models and applications. In *EG State of the Arts Report*, pages 113–136, 2010.

[8] J. Wilhelms and A. Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.