ACM Symposium on Solid Modeling and Applications (2004)
G. Elber, N. Patrikalakis, P. Brunet (Editors)

A Framework for Multi-dimensional Adaptive Subdivision Objects

Yu-Sung Chang and Hong Qin[†]

State University of New York at Stony Brook

Abstract

Despite the growing interest in subdivision surfaces within the computer graphics and geometric processing communities, subdivision approaches have been receiving much less attention in solid modeling. This paper presents a powerful new framework for a subdivision scheme that is defined over a simplicial complex in any n-D space. We first present a series of definitions to facilitate topological inquiries during the subdivision process. The scheme is derived from the double $(k+1)$-directional box splines over k-simplicial domains. Thus, it guarantees a certain level of smoothness in the limit on a regular mesh. The subdivision rules are modified by spatial averaging to guarantee C^1 smoothness near extraordinary cases. Within a single framework, we combine the subdivision rules that can produce 1-, 2-, and 3-manifold in arbitrary n-D space. Possible solutions for non-manifold regions between the manifolds with different dimensions are suggested as a form of selective subdivision rules according to user preference. We briefly describe the subdivision matrix analysis to ensure a reasonable smoothness across extraordinary topologies, and empirical results support our assumption. In addition, through modifications, we show that the scheme can easily represent objects with singularities, such as cusps, creases, or corners. We further develop local adaptive refinement rules that can achieve level-of-detail control for hierarchical modeling. Our implementation is based on the topological properties of a simplicial domain. Therefore, it is flexible and extendable. We also develop a solid modeling system founded on our theoretical framework to show potential benefits of our work in industrial design, geometric processing, and other applications.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – *Curve, surface, solid, and object representations*

1. Introduction

Since Requicha and Voelcker [RV82]'s famous survey paper in 1982, the past two decades have witnessed significant growth in solid modeling, especially in the development of new solid representation techniques. In essence, we can classify the existing techniques by how they represent models: namely, either continuous or discrete representation. Parametric representations and implicit function methods are two classic examples of the continuous representation. Although models will be eventually approximated to vertices, edges, and/or faces to be displayed on computer screen, they are described as an image of continuous functions, level-sets of functions, or patches of locally smooth functions, internally. As Boehm *et al.*[BFK84] surveyed, parametric curves and surfaces had been widely used especially in computer-aided design and manufacturing for a long time. Bernstein-Bézier solids [Las85], B-spline solids, and other tensor product based [GP89] approaches are typical examples of parametric representations in solid modeling. Implicit function methods, such as CSG [PS94] and blobby models [WMW86], define an object by a solution set of implicit functions. In this method, it is especially easy to perform set operations, such as intersections and unions. However, even for the implicit function method, which has a great flexibility in the topologies of the models that it can represent, it is relatively hard to model objects with different dimensionality (*e.g.*, non-manifold objects) in a single representation.

In contrast, the discrete representations include cell decomposition, triangular models for surfaces, and tetrahedral or hexahedral models for solids. These techniques represent models as a finite number of elements, such as pixels, voxels, triangles, or tetrahedra. Because there is no function involved, it can represent an object with arbitrary manifold properties, such as a combination of lines and surfaces, self-intersecting faces, *etc*. We can achieve a certain level of detail, for instance, by using an octree or a progressive mesh [Hop96]. One obvious problem with the discrete representation is that there is no geometric interpretation on the elements. Thus, we have to rely on approximation to obtain any geometric properties. Topological inquiry is rather easier, but it still requires intensive graph searching in most cases.

In fact, there is no clear distinction between these two categories in current modeling applications. Every continuous representation has to be converted to discrete objects for computer display and practical use, and sophisticated approximation techniques

[†] Email: {yusung|qin}@cs.sunysb.edu

have been developed to obtain geometric information from discrete models. Accordingly, various computer-aided design systems utilize both representations in a hybrid fashion. The subdivision technique is an example of new representation that shares the features of both categories. From an initial control mesh that is essentially discrete, we successively perform a series of computations – mostly simple linear combinations – to obtain the next level of mesh that is finer than the previous one. In the limit, we end up getting an object which is an image of smooth functions. Topological information can be acquired from the initial meshes, whereas geometrical properties can be obtained from the subdivision matrix analysis.

In this paper, we establish a framework that is based on flexible parametric domains and powerful subdivision rules which can be applied to objects with complicated dimensionality. The goals of our new approach are as follows: (1) Define a parametric domain that provides high flexibility in modeling and simplicity in topological inquiry; (2) Represent objects with multiple dimensions in a single framework; (3) Develop subdivision rules for arbitrary manifolds and multiple dimensions; and (4) Support features and level-of-detail (LODs) control. We begin the discussion by reviewing the previous work that is related to the goals specified above.

Parametric domain. For nearly all subdivision schemes, the tensor-product is a standard way to expand the dimensions. For instance, the Catmull-Clark scheme by Catmull and Clark [CC78] and the Multi-linear Cell Averaging scheme by Bajaj *et al.*[BSWX02] both utilize tensor-product cubic B-splines. In any case, their parametric domains should have the form of a tensor-product space. Some shortfalls are apparent for the tensor-product space. First, tensor-product functions have a higher polynomial degree than the functions that are natively defined over the space with the same smoothness. Secondly, tensor-product meshes are less flexible than others, such as triangular or tetrahedral meshes. Also, there is an ambiguity problem if each face of the mesh is not planar. We choose a simplicial mesh as our parametric domain for the framework because of its flexibility, extendability, and the ability to accommodate non-manifolds. There has been substantial research on simplicial meshes. For instance, Floriani *et al.*[FMPS02, FMP03] proposed techniques to represent progressive non-manifolds by simplicial meshes. Most of the work on simplicial meshes has been related to numerical analysis, especially for the finite element method (FEM).

Subdivision schemes. Since one of the purposes of the framework is to represent multi-dimensional objects, we are required to have subdivision schemes that can be easily extended to various dimensions. Moreover, as explained in the previous paragraph, we want the schemes to be based on a simplicial domain. Cubic B-spline subdivision is one of the simplest schemes for curves. An example of the surface subdivision schemes that are based on 2-simplices, or triangular meshes, is Loop's scheme [Loo87]. For 3-D solid objects, MacCracken and Joy [MJ96] proposed the tensor-product extension of the Catmull-Clark subdivision in the volumetric setting, mainly for the purpose of free-form deformation in 3-D space. Later on, Bajaj *et al.* [BSWX02] further extended the scheme with an analysis based on numerical experiments. They are both the tensor-product extensions of the cubic B-spline curves, and hence, are not suitable for our purpose. Most recently, Chang *et al.*[CMQ02] suggested a non-tensor-product based subdivision scheme over simplicial meshes whose limit converges to the trivariate box spline. They also proposed an interpolatory subdivision solid scheme [CMQ03] over simplicial complexes. In fact, the cubic B-spline scheme, Loop's scheme, and Chang's box spline solid scheme are the direct analogs of the double directional box splines over 1-, 2-, and 3-simplicial meshes. These three schemes serve as basic rules for our framework. In addition, even for a single dimensional scheme, non-manifold regions can occur through self-intersection. Ying *et al.*[YZ01] suggested modified rules for the Loop's scheme to deal with non-manifold surfaces. In our framework, the cases are more complex than those of a single subdivision scheme.

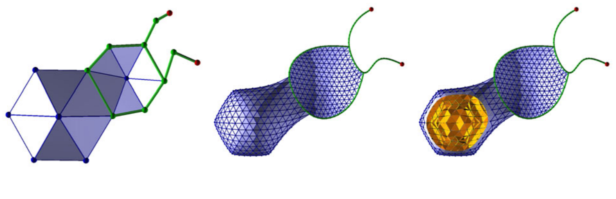

Figure 1: *A non-manifold object represented by the subdivision. (a) The initial complex that consists of 1-, 2-, and 3-simplices. (b) After level 3. (c) The cross-section of the 3-manifold reveals the internal structure.*

Non-manifolds, features and detail control. The models represented by subdivision schemes tend to be smooth everywhere. However, the vast majority of real-world models, especially manufactured objects, have sharp features. Hoppe *et al.*[HDD*94] proposed modifications to Loop's scheme to represent features like corners and creases. We follow similar approaches to introduce features within the framework. For level-of-detail control, a considerable amount of research has been done for progressive mesh approaches. For instance, Popovic *et al.*[PH97] presented the idea of a progressive simplicial complex. In our framework, we follow the traditional local refinement method for triangular and tetrahedral meshes to achieve the LODs.

The rest of the paper is organized in the following fashion. In Section 2, we define a parametric domain and document other topological definitions, which serve as the fundamentals of our unique framework. In Section 3, we discuss the subdivision rules, their modifications, and a brief sketch of the analysis. We tackle the problem of features and level-of-detail control in Section 4. Section 5 describes the implementation of the framework in detail and demonstrates several models generated by our framework. Finally, we discuss future work and conclude the paper in Section 6.

2. Simplicial Complex

In the paper, we define an object in the space as a manifold, or a union of manifolds. Topologically, a manifold is defined as a locally Euclidean countable Hausdorff space. By locally Euclidean, we mean that for any point x on the manifold, we can find a homeomorphic map from an open subset of $\mathbb{R}^n$. In addition, there is a manifold with boundary if the domain of a local Euclidean map is half-space-like. From the solid modeling point of view, it is a matter of choosing a continuous, injective, and surjective function from an appropriate domain in Euclidean space.

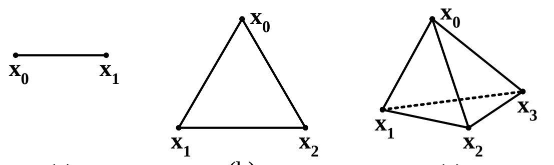

Figure 2: *Examples of simplices. (a) A 1-simplex, (b) a 2-simplex, and (c) a 3-simplex.*

Throughout the framework, we choose the domain to be k-simplices in $\mathbb{R}^3$ ($k \leq 3$). Local Euclidean maps are defined and evaluated by a series of subdivision rules whose supports are limited in a single simplex or a small number of adjacent simplices. In fact, the initial control points for the subdivision rules also provide the simplicial domain of our objects. Moreover, they are not only homeomorphic, but also satisfy the higher-level of smoothness on their supports and are C^1 across the simplices. In the next few sections, we introduce several definitions related to the simplicial complex that are to be utilized for various topological inquiries during the subdivision process.

2.1. Definitions

Our domain of choice is a simplicial complex in $\mathbb{R}^n$. A k-simplex S can be defined as a set in $\mathbb{R}^n$,

$$S = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} = \sum_{i=1}^{k} c_i(\mathbf{x}_i - \mathbf{x}_0)\}, \tag{1}$$

where

$$c_i \geq 1, \ \sum_{i=1}^{k} c_i = 1, \ \mathbf{x}_i \in \mathbb{R}^n. \tag{2}$$

Since S can be uniquely determined by $k+1$ points $\mathbf{x}_0$, $\mathbf{x}_1$, ... ,$\mathbf{x}_k$, and is independent of their ordering, we simply use a set notation $S := \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_k\}$. In this paper, we limit k to be less than or equal to three. Note that any subset of S also forms a simplex. Geometrically, each subset can be considered as a face, an edge, or a vertex. We call k the *dimension* of the simplex S, or $\dim(S)$.

In any collection of simplices, we call a simplex a *subsimplex* if it is a subset of any other member of the collection. Likewise, it is called a *proper subsimplex* if it is a proper subset of a simplex in the collection.

A simplicial complex, or a complex, $\mathcal{C}$ is a collection of simplices where: (1) the subsimplices of each simplex in $\mathcal{C}$ is in $\mathcal{C}$; (2) the intersection of any two simplices of $\mathcal{C}$ is a subsimplex of both. The second property prevents the introduction of T-junctions or the improper incursion among simplices. Also, a nonempty subset $\mathcal{D}$ of a simplicial complex $\mathcal{C}$ is called a *simplicial subcomplex* if it also satisfies the properties. We simply call it a *subcomplex*. The dimension of a complex is defined by the highest dimension of simplices in it.

In summary, the domain space of our framework can be expressed as the pair of the following sets:

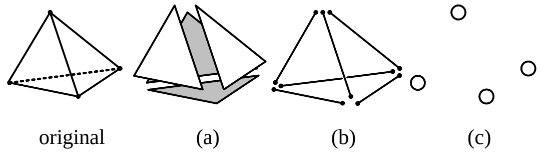

Figure 3: *The subsimplices of a 3-simplex. (a) The 2-subsimplices, (b) the 1-subsimplices, and (c) the 0-subsimplices.*

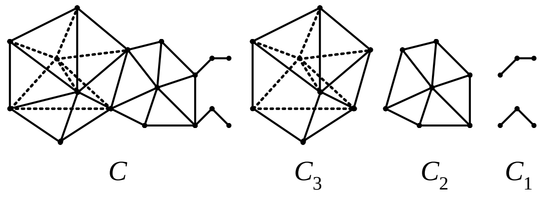

Figure 4: *Complex decomposition. A complex $\mathcal{C}$ can be decomposed into $\mathcal{C}_k$'s with $k = 1, 2, 3$.*

- Set of vertices

$$\mathcal{V} = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^3\}, \tag{3}$$

- A simplicial complex

$$\mathcal{C} = \{S \subset \mathcal{V} \mid S \neq \emptyset, \ |S| \leq 4\}, \tag{4}$$

with the following property:

$$\text{If } S \in \mathcal{C}, \text{ then } T \in \mathcal{C} \text{ for all } T \subset S, \ T \neq \emptyset. \tag{5}$$

2.2. Complex Decomposition

A complex $\mathcal{C}$ can contain simplices of different dimensions. Since each k-simplex is to be used as a part of the initial control points of a k-manifold, we need to decompose $\mathcal{C}$ with respect to the dimensions of the simplices. We define $\mathcal{C}_k$ as the largest subcomplex of $\mathcal{C}$, whose maximal elements have the dimension k. We consider the maximality by set inclusion order. In other words, $\mathcal{C}_k$ comprises of all maximal k-simplices and their subsimplices in $\mathcal{C}$. We call it a *k-subcomplex*. Therefore, we can express $\mathcal{C}$ as:

- k-subcomplex decomposition

$$\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3, \tag{6}$$

where each $\mathcal{C}_k$ satisfies the following property:

$$\text{If } S \in \mathcal{C}_k \text{ and is maximal in } \mathcal{C}_k, \text{ then } \dim(S) = k. \tag{7}$$

In Section 3, we define k-manifolds (with boundary) over the k-subcomplex using appropriate subdivision rules. However, $\mathcal{C}_k$'s are not mutually exclusive. This fact leads us to the need for special rules across the intersections of the k-subcomplexes. In fact, the intersections represent non-manifold regions in the result. Moreover, some non-manifold regions could appear within $\mathcal{C}_1$ and $\mathcal{C}_2$, since the complex is defined over $\mathbb{R}^3$.

2.3. Boundary and Non-manifold Simplex

A face of a k-simplex S is simply defined as a $(k-1)$-subsimplex of S. A boundary of a complex can be defined as follows:

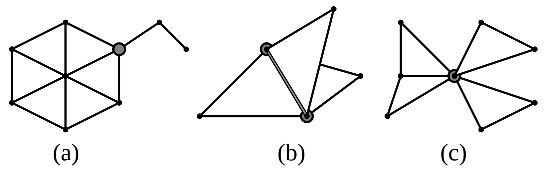

Figure 5: *Examples of complexes containing non-manifold simplices. (a) Type 1, (b) type 2, and (c) type 3. The vertices or edges in gray are the non-manifold simplices.*

- Boundary simplex: If $(k-1)$-simplex $S \in \mathcal{C}$ is a face of a maximal k-simplex, and is not a subsimplex of any other simplices, than S defines a boundary. We call it a k-boundary simplex.

It is clear that boundary simplices and their subsimplices form a subcomplex of $\mathcal{C}$. It is denoted by $\partial\mathcal{C}$.

It is not possible to define a manifold map over a certain region. For instance, if a 1-manifold and a 2-manifold meet at a single vertex, the vertex is not locally Euclidean, even though we can define each manifold over a 1- and a 2-simplex, respectively. On the other hand, if a 1-manifold intersects itself, it is not possible to find the map either. We categorize the non-Euclidean part of our domain complex as follows:

- Non-manifold simplex: A k-simplex $S \in \mathcal{C}$ is a non-manifold simplex, if
  1. $S \in \mathcal{C}_k \cap \mathcal{C}_l$ where $k \neq l$.
  2. $S \in \mathcal{C}_k$ exclusively, and it is a face to more than two k-simplices, or
  3. $S \in \mathcal{C}_k$ exclusively, and it is a face to more than one k-boundary simplex.

We call them a type 1, a type 2 and a type 3 non-manifold simplex, respectively. We employ various strategies to tackle the non-manifold cases. Generally, non-manifold simplices create ill-posed problems. There could be several different solutions to meet a particular requirement in certain applications. We rely on a user-specific preference to resolve the problems. If no rule is specified by the user, we use the subdivision rules for 3-manifolds to spatially blend the manifolds of different dimensions. It is worthwhile to mention that the type 2 only occurs in $\mathcal{C}_1$ and $\mathcal{C}_2$ because our complex is defined in $\mathbb{R}^3$.

3. Subdivision Scheme

In the previous section we defined the domain of the framework as a simplicial complex. Our object can be represented by the sum of smooth basis functions that are defined locally over the simplices in the complex:

$$f(\mathbf{x}) = \sum \mathbf{p} N(\mathbf{x}), \tag{8}$$

where $\mathbf{p} \in S \in \mathcal{C}$ with $\dim(S) = 1$. Therefore, the 1-simplices (or vertices) in the complex act as the control points of the shape. $N(\mathbf{x})$ is a basis function with local support defined over the complex. Basis functions form a partition of unity on $\mathcal{C}$. We choose the box spline as the function $N(\mathbf{x})$ whose support lies in the 1-ring of simplices. For multivariate cases, we do *not* use the tensor-product generalization of splines in strong contrast to many other subdivision schemes, since our domain is based on a complex. Instead, we introduce multivariate box splines with simplex support. One example is Loop's scheme [Loo87] for surfaces. For 3-D, we use the box spline solid that has been employed in our previous work [CMQ02]. Non tensor-product box splines are particulary useful in the subdivision process, since: (1) Their subdivision rules are obtained intuitively from their definitions; (2) They can achieve comparable smoothness with relatively low polynomial degree.

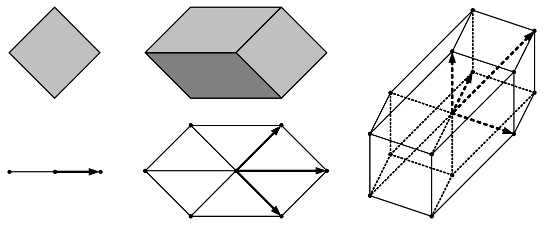

Figure 6: *The domain support for the box splines. The upper images are the unit cubes whose projections are taken. The thick arrows are the direction vectors. For (c), we only display the support, since it is hard to visualize a 4-hypercube.*

3.1. Box Splines

Box splines can be understood as projections of hypercubes into $\mathbb{R}^n$. Because of this, each box spline $N_D(\mathbf{x})$ can be represented by the collection of direction vectors $D = [\delta_1, \ldots, \delta_d]$. Note that each $\delta_i \in \mathbb{R}^k$ is the projection of an edge of a hypercube, and thus, is not necessarily distinct. We employ the *double* $(k+1)$*-directional* box spline for each k-manifold defined over $\mathcal{C}_k$, except $k=0$. Each double $(k+1)$-directional box spline has the properties as follows:

1. For $k=1$, the direction vectors are chosen to be $D = [1,1,1,1]$, where each 1 is a unit vector lying in a 1-simplex, or a line segment. It is double 2-directional, but the two directions coincides in a 1-simplex. In fact, this is exactly the same spline as the cubic B-spline. As such, it follows the same properties as cubic B-splines.
2. For $k=2$, $D = [(1,0),(1,0),(0,1),(0,1),(1,1),(1,1)]$. The box spline N_D is the double 3-directional box spline. As shown in Figure 6(b), its domain lies in the 1-ring of 2-simplices, or triangles. Loop's scheme is based on this box spline.
3. For $k=3$, $D = [\mathbf{e}_1,\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_2,\mathbf{e}_3,\mathbf{e}_3,\mathbf{u},\mathbf{u}]$, where $\mathbf{e}_k$ is a unit vector for each axis in $\mathbb{R}^3$ and $\mathbf{u} = \sum \mathbf{e}_k$. The support of the box spline is shown in Figure 6(c). Unfortunately, it is not embedded in the 1-ring of 3-simplices, or tetrahedra. However,by adding few more edges, we can turn it into a simplicial complex.

Generally, the box splines satisfy the following two properties, as proven in [dBHR93]:

1. The box spline N_D is piecewise polynomial of degree $|D|-k$.
2. The box spline N_D is a C^m function where $m = |D| - |D'| - 2$, and D' is a maximal subset of D that does not span $\mathbb{R}^k$.

For instance, the double $(k+1)$-directional box splines are piecewise polynomials of degree $k+2$.

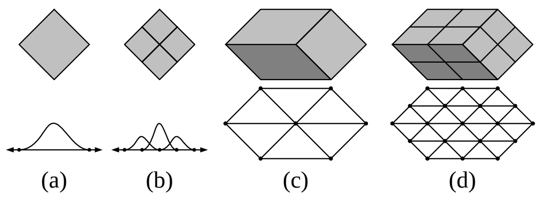

Figure 7: *Subdivision of the box splines.*

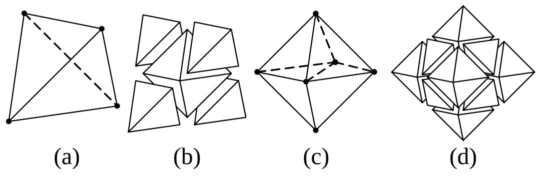

Figure 8: *Split a tetrahedron and an octahedron.*

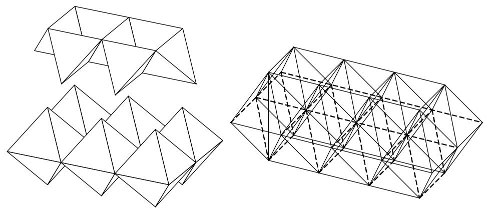

Figure 9: *Octet-truss.*

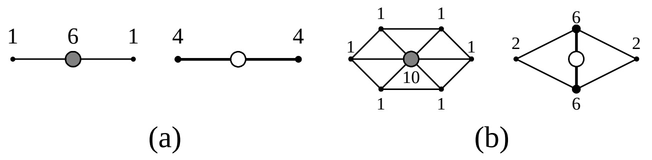

Figure 10: *Regular subdivision rules. (a) The 1-simplex rules. (b) The 2-simplex rules.*

3.2. Subdivision Meshes

The box splines can be expressed as a sum of the box splines with the half-sized supports (See Figure 7). Using this property, we can find out the rules for the subdivision scheme. We first consider the split of the domain. As mentioned in the previous sections, our box splines are defined over the 1-ring of k-simplices. It is easy to subdivide the domain if it is comprised of only 1-, or 2-simplices as shown in Figure 7(b) and (d). Trivial edge bisection results in the half-sized simplices of the originals in these cases. However, it is not so simple for 3-simplices. A 3-simplex, or a tetrahedron, does not split into congruent tetrahedra by edge bisecting. In fact, there is no way to obtain congruent tetrahedra from any subdivision of a tetrahedron. This is also related to the problem that a single type of tetrahedra can not fill the entire $\mathbb{R}^3$, unlike 2-simplices, or triangles in $\mathbb{R}^2$. We resolve the problem by the following approaches:

1. The boundary of the projection of a 4-hypercube on $\mathbb{R}^3$ (See Figure 6(c)) is a rhombic dodecahedron. It is well-known that this polytope can fill the space.
2. By introducing a few additional edges, we can decompose the dodecahedron into several tetrahedra.
3. A single tetrahedra can be split into four congruent tetrahedra and one octahedron, as shown in Figure 8(b). Also, an octahedron can be split into eight tetrahedra and six congruent octahedra (See Figure 8(d)).
4. If we keep continuing this process, then we get a semi-regular space-filling structure called *octet-truss* (See Figure 9). It is not difficult to figure out that the simplicial split of the dodecahedron can be embedded in the truss, and thus can provide us the subdivision of the 3-simplex domain.
5. We store one diagonal inside an octahedron, as shown in Figure 8(c), to keep track of the adjacency of each vertex. In fact, each octahedron can be considered as a family of four tetrahedra.

3.3. Regular Subdivision Rules

Even though it is possible to figure out the subdivision rules using the definitions of the box splines, it is more convenient to use the generating functions of the box splines and their recursive relations. It is known that the coefficients of the generating functions can provide us the coefficients for the subdivision rules, as proven in [WW01]. In general, the generating function $S_D(\mathbf{z})$ for the box spline $N_D(\mathbf{x})$ can be expressed as:

$$S_D(\mathbf{z}) = \frac{1}{2^{d-k}} \prod_{i=1}^{d} (1+\mathbf{z}^{\delta_i}), \tag{9}$$

where $d = |D|$. Note that the power of $\mathbf{z}$ follows the multi-index notation. For each k, the generating functions of the double $(k+1)$-directional box splines are:

- $k = 1$:
$$S_D(z) = \frac{1}{8}(1+z)^4. \tag{10}$$
- $k = 2$:
$$S_D(z_1,z_2) = \frac{1}{16}(1+z_1)^2(1+z_2)^2(1+z_1z_2)^2. \tag{11}$$
- $k = 3$:
$$S_D(z_1,z_2,z_3) = \frac{1}{32}(1+z_1)^2(1+z_2)^2(1+z_3)^2(1+z_1z_2z_3)^2. \tag{12}$$

We can find the subdivision rules for the regular simplicial meshes by assigning the coefficients of the $\mathbf{z}^{\delta_i}$'s to the vertex with the coordinates δ_i. We can summarize the rules as follows:

- Regular k-simplex subdivision rules:

Vertex points (for each vertex $\mathbf{x}_i$):

$$\mathbf{v}_{new} = \frac{1}{2^{k+2}} \Big\{ (2^{k+1}+2)\mathbf{x}_i + \sum_{\mathbf{x}_j \in \rho(\mathbf{x}_i)} \mathbf{x}_j \Big\}. \tag{13}$$

Edge points (for each edge $\mathbf{e}_i = [\mathbf{x}_i, \mathbf{x}_{i+1}]$):

$$\mathbf{e}_{new} = \frac{1}{2^{k+1}} \Big\{ (2^{k-1}+1)(\mathbf{x}_i + \mathbf{x}_{i+1}) + \sum_{\mathbf{x}_j \in \rho(\mathbf{e}_i)} \mathbf{x}_j \Big\}. \tag{14}$$

Cell points (for each octahedral cell $[\mathbf{x}_i, \ldots, \mathbf{x}_{i+3}, \mathbf{x}_j, \mathbf{x}_{j+1}]$, with the diagonal $[\mathbf{x}_j, \mathbf{x}_{j+1}]$):

$$\mathbf{c}_{new} = \frac{1}{8} \Big\{ (\mathbf{x}_i + \cdots + \mathbf{x}_{i+3}) + 2(\mathbf{x}_j + \mathbf{x}_{j+1}) \Big\}. \tag{15}$$

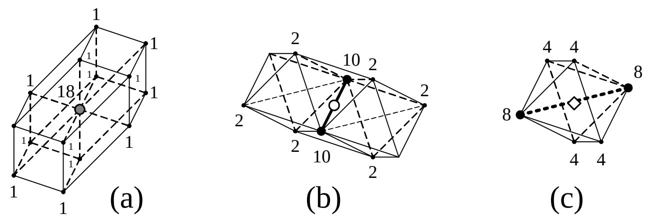

Figure 11: *Regular 3-simplex subdivision rules. (a) Vertex point, (b) edge point, and (c) cell point rules.*

Here, we use more conventional names for 0-, 1-, 2-, and 3-simplices, namely, vertices, edges, and cells, respectively. $\rho(\cdot)$ denotes the 1-ring of neighboring vertices of a vertex or an edge. In the regular k-simplicial meshes, $|\rho(\mathbf{x})| = 2^{k+1} - 2$, and $|\rho(\mathbf{e})| = 2^k - 2$ for each vertex $\mathbf{x}$ or edge $\mathbf{e}$. Note that each k-manifold generated by the subdivision rules on the regular mesh satisfies C^k smoothness as mentioned above.

3.4. Extraordinary Subdivision Rules

In practice, a complex $\mathcal{C}$ could contain a vertex or an edge, that does not have a regular number of neighbors $|\rho(\cdot)|$ (or *valences* for vertices). We call them the *extraordinary* cases. They require modified rules to accommodate the lack (or the excessiveness) of neighbors. Fortunately, the extraordinary cases are isolated over the subdivision processes. Also, some of the regular rules do not require any extraordinary rule. For instance, the 1-simplex rules do not have any extraordinary case. For the 2-simplex rules, there could be only extraordinary vertices. Likewise, no extraordinary cell point rule is required for the 3-simplex rules.

The extraordinary vertex rule for a 2-simplex has been well studied and there is a considerable amount of literature suggesting the coefficients for the rule that guarantee at least C^1 smoothness in the limit. For instance, the original Loop scheme [Loo87] suggests the coefficients for a vertex with valence m that are derived from the discrete Fourier analysis and the eigenvalue analysis of the subdivision matrix. We adopt the values proposed by Warren *et al.*[WW01]:

- Modified 2-simplex subdivision rules:

Vertex points $(|\rho(\mathbf{x}_i)| = m)$:

$$\mathbf{v}_{new} = (1-mc)\mathbf{x}_i + c \sum_{\mathbf{x}_j \in \rho(\mathbf{x}_i)} \mathbf{x}_j, \qquad (16)$$

where $c = \frac{3}{16}$ for $m = 3$, $c = \frac{3}{8m}$, otherwise.

Similar modifications are required for the 3-simplex subdivision rules:

- Modified 3-simplex subdivision rules:

Vertex points $(|\rho(\mathbf{x}_i)| = m)$:

$$\mathbf{v}_{new} = \frac{9}{16}\mathbf{x}_i + \frac{7}{16m} \sum_{\mathbf{x}_j \in \rho(\mathbf{x}_i)} \mathbf{x}_j. \qquad (17)$$

Edge points $(|\rho(\mathbf{e}_i)| = m)$:

$$\mathbf{e}_{new} = \frac{5}{16}(\mathbf{x}_i + \mathbf{x}_{i+1}) + \frac{3}{8m} \sum_{\mathbf{x}_j \in \rho(\mathbf{e}_i)} \mathbf{x}_j. \qquad (18)$$

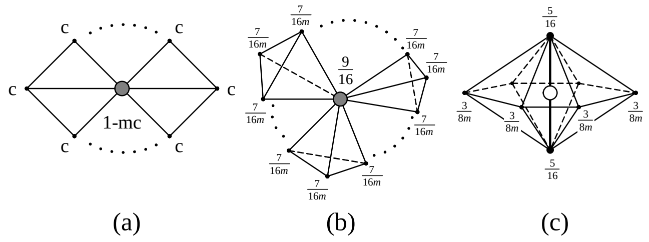

Figure 12: *Modified k-simplex subdivision rules.*

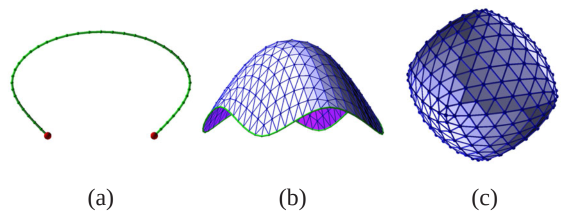

Figure 13: *Examples of manifolds with boundary. (a) A 1-manifold with boundary. (b) A 2-manifold with boundary. (b) A 3-manifold with boundary.*

3.5. Manifold with Boundary and Non-manifold Region

The boundaries of k-manifolds cannot be represented by the k-simplex subdivision rules, because they are defined by the faces of k -simplices. Instead, we use the $(k-1)$-simplex subdivision rules to represent the boundaries. Since all of the subdivision rules rely only on the 1-rings of neighbors, this approach causes no additional trouble between the boundary and the interior simplices. It is, in fact, a standard approach for most subdivision surface schemes. Figure 13 demonstrates examples of such boundary cases.

Some regions of the complex require special rules because they cannot serve as the domain of manifolds. We categorize the cases into three types, as explained in Section 2.3. In each case, we rely on user input to determine which rules to apply. If the user has not provided a choice, we try to find the best possible way to deal with it. Ying *et al.*[YZ01] proposed detailed approaches to overcome non-manifold topology with subdivision surfaces. They involve a the specially modified Loop's scheme and a geometric fitting process. Since our domain is in $\mathbb{R}^3$ and we have the 3-simplex subdivision rules that can accept an arbitrary manifold with lower dimension, our solutions could be much simpler, as described below.

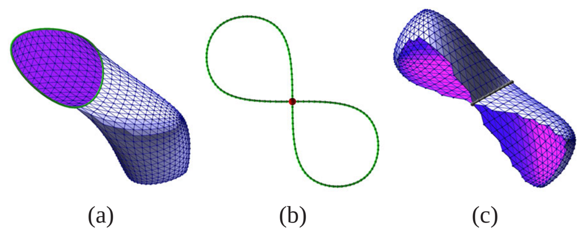

Figure 14: *Examples of non-manifold cases. (a) A type 1 case by the 2- and 3-manifold intersection. (b) A type 2 case by a single 1-manifold. (c) The cross-section of the type 2 case.*

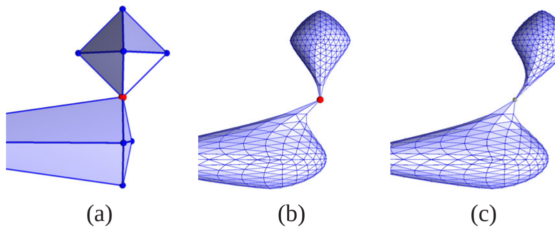

Figure 15: *Type 3 non-manifold rules. (a) The initial complex. (b) The subdivision by Rule G-1. (c) The subdivision by Rule G-2. The red vertex preserves its position in (b), while it is blended in (c).*

Recall that the dimension of a non-manifold simplex is either zero (a vertex), or one (an edge):

- Type 1 is a region where the manifolds with different dimensions meet. In this case, we can follow the subdivision rules for one k-simplex of our choice. The others ignore the region (no rule applied).
- Type 2 is a region where a multiple manifold of a single dimension intersects. This region can be considered as a self-intersection. One possible solution is to choose one pair of the simplices on which we apply the subdivision rule. The others ignore the region (no rule applied).
- Type 3 is a region where the boundaries of multiple manifolds of a single dimension coincide. There is no specific solution for this case, except the general rules below.
- Regardless of the type, we can apply one of the general rules as follows:

  Rule G-1 Treat the intersection as a 0-, or 1-singularity.
  Rule G-2 Use the 3-simplex subdivision rules with the relief topology condition (only consider connectivity).

By a subdivision rule with the relief topology condition, we mean that the rule only considers the connectivity between each vertex when acquiring the 1-ring of neighbors, regardless of the existence of a simplex in between. Figure 14 and 15 illustrate examples of non-manifold cases. In Figure 14(a), a 2-mainfold (the purple area) intersects with a 3-manifold (the blue area). Therefore, it forms a type 1 case. For this particular case, a user has decided to follow the 2-manifold rule. Thus, the intersected area follows the 2-manifold boundary rule, and the 3-manifold is blended into it. Figure 14(b) and (c) show typical type 2 cases. In Figure 14(b), the 1-manifold intersects itself at a point (the red vertex). A multiple number of surfaces intersects at an edge in Figure 14(c). For both cases, we use Rule G-2 to blend non-manifold parts into the bodies. Figure 15 shows the effects of the different rules. In Figure 15(b), the user selects the red vertex to be a singularity (Rule G-1). Hence, we only apply the 0-mask (*i.e.*, the 1×1 identity matrix) on the vertex during the subdivision process. Thus, it preserves the position during the subdivision. However, in Figure 15(c), we follow Rule G-2. As a result, the vertex has been moved according to the positions of the 1-ring neighbors because we use the subdivision rules for 3-simplices. In the end, the final shape is much smoother and all the boundaries are well blended. We should mention that the suggested rules do not represent all the possible solutions. Nonetheless, we can introduce a new rule depending on the requirement of a particular application.

3.6. Analysis

Smoothness analysis is required only for the extraordinary cases, since the regular rules are based on the recursive property of the box splines and the generating functions. The convergence and smoothness of the regular cases are well-documented in [dBHR93] and [WW01]. For the 1-simplex rules, there is no extraordinary case, and thus, no extraordinary analysis is required. The 2-simplex rules require analysis of the extraordinary vertex case. This analysis, based on the spectral analysis technique, has been proposed by many researchers. For instance, Micchelli [MP87], Prautzsch [Pra85, Pra98], Reif [Rei95b, Rei95a], and more recently, Zorin [ZSS96, Zor00] investigated the sufficient and necessary conditions of convergence and the C^1 smoothness. Given the abundance of the related literature and in the interest of space, we only sketch the idea in brief.

Since the subdivision process is a linear combination, in essence, we can represent the rules locally by the subdivision matrix **S**,

$$\mathbf{p}^{\ell+1} = \mathbf{S}\mathbf{p}^{\ell}, \tag{19}$$

where $\mathbf{p}^{\ell}$ consists of a vertex $\mathbf{x}^{\ell}$ at the subdivision level ℓ and its neighbors $\mathbf{x}^{\ell} = [\mathbf{x}_1^{\ell}, \ldots, \mathbf{x}_m^{\ell}]$. We assume that λ_i's are the (left) eigenvalues of **S** in non-increasing order. If the set of the initial vertices $\mathbf{p}^0$ is expressed by the corresponding eigenvectors $\mathbf{v}_i$ in the eigenspace of the matrix **S**,

$$\mathbf{p}^0 = \mathbf{a}_0\mathbf{v}_0 + \mathbf{a}_1\mathbf{v}_1 + \cdots + \mathbf{a}_n\mathbf{v}_n, \tag{20}$$

the limit process can be expressed as

$$\mathbf{p}^{\ell} = \mathbf{S}^{\ell}\mathbf{p}^0 = \lambda_0^{\ell}\mathbf{a}_0\mathbf{v}_0 + \lambda_1^{\ell}\mathbf{a}_1\mathbf{v}_1 + \cdots + \lambda_n^{\ell}\mathbf{a}_n\mathbf{v}_n. \tag{21}$$

Hence, the limit position $\mathbf{x}^{\infty}$ of $\mathbf{x}^0$ can be expressed by,

$$\mathbf{x}^{\infty} = \frac{\lambda_0\mathbf{x}_1^0 + \cdots + \lambda_m\mathbf{x}_m^0}{\lambda_0 + \cdots + \lambda_m}, \tag{22}$$

under the condition:

$$\lambda_0 = 1 > \lambda_1 \geq \lambda_2 > \lambda_3, \cdots, \lambda_n. \tag{23}$$

As shown in Figure 12(a), the matrix **S** has a cyclic structure due to its planar symmetry in the 2-simplex case. Therefore, after reordering $\mathbf{p}^0$, it is possible to apply the discrete Fourier transform on **S** to obtain the closed form of the eigenvalues. Combined with the condition (23), this leads us to the coefficients of the subdivision rule (16). Accompanying analysis on the characteristic map suggested by Reif [Rei95b] can guarantee the C^1 smoothness around the vertex.

The same process can be applied to the 3-simplex edge case to prove C^1 smoothness, because it still carries the planar symmetry in **S**. However, the subdivision matrix for the 3-simplex vertex rule does not have any symmetry at all in general. This results in the failure of the application of the discrete Fourier transform, and only a numerical process can be employed to acquire the eigenvalues. In fact, Bajaj *et al.*[BSWX02] suggested the condition for C^1 smoothness of three dimensional case as:

$$\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq \lambda_3 > \lambda_4, \cdots, \lambda_n, \tag{24}$$

through their empirical analysis. Our suggested values are based on the spatial averaging and the assumption of the even distribution of the valences of the vertices in the 1-ring planar graph. The eigenvalues for our subdivision matrix have confirmed that the condition

(24) is satisfied for most of the cases, except very small $|\rho(\cdot)|$ (valence 4, in particular) which hardly occurs in any real applications. Nonetheless, our experiments show that there are no visible degenerations of the 3-manifold even after the very large number of the subdivision processes. More in-depth analysis on the 3-D cases is not yet fully explored, and it will be the subject of our future publication.

4. Singularity and Adaptivity

Even though the subdivision rules that we have presented so far are ideal for representing smooth objects, it is desirable to have a model with sharp features, such as cusps, creases, or corners, especially in real-world applications. Also, we may want to have more details in some part of the model without subdividing the whole complex. In the following sections, we discuss the extensions of the framework that can increase its benefit in practical solid modeling.

4.1. Singularity Representation

Hoppe *et al.*[HDD*94] suggested a modification of Loop's scheme to represent sharp features within smooth surfaces. Our basic idea is similar to theirs. However, we generalize the approach to apply to multi-dimensional models.

A manifold defined by the subdivision rules is C^1 smooth over the complex $\mathcal{C}$ except in non-manifold regions. To represent features within the manifold: (1) We need to specify the area of the domain where the features occur; (2) We need to specify the subdivision rules to represent the features in the manifold. Among many types of features, we only consider "sharp" features, where the manifold is continuous, but is *not* differentiable. We call this type of features a *singularity* for convenience. We define a *k-singular* simplex by:

- *k*-singular simplex: A *k*-simplex $S \in \mathcal{C}$ is a *k*-singular simplex, if and only if: (1) There exists no C^1 map to *l*-manifolds defined over any simplex $T \in \mathcal{C}$, where $S \subset T$ and $k < l$. (2) It is possible to define a differentiable map on the singular simplex to *k*-manifolds.

We consider a subcomplex $\mathcal{S} \subset \mathcal{C}$, which is a collection of all singular simplices in $\mathcal{C}$. Since they are a complex by themselves, all definitions and subdivision rules that are applied to the complex $\mathcal{C}$ are also applicable to $\mathcal{S}$. Basically, $\mathcal{S}$ generates embedded manifolds within the original manifolds on $\mathcal{C}$. When applying the subdivision rules, if a vertex $\mathbf{x}$ or an edge $\mathbf{e}$ belongs to a maximal simplex in $\mathcal{S}$, we only follow the subdivision rules that match the dimension of the simplex, and ignore any other simplices that may contain the singular simplex. Figure 16 illustrates examples of singularities which our framework can represent. As shown in Figure 16(a), if a vertex (a 1-simplex) is assigned to be singular, then the scheme only applies the 0-mask on the vertex during the subdivision. Therefore, the vertex does not change its position at each subdivision level. However, other vertices around it follow the normal rules. As a result, we can obtain an object which is smooth except at one singular vertex and in its local area. This singularity is particularly useful to generate a cusp on the part of a manifold. In Figure 16(b), a user has assigned one vertex and all edges that go through it as singular. The 0-mask is applied to the vertex, and each edge follows the 1-simplex edge rule. It effectively produces a corner and three creases

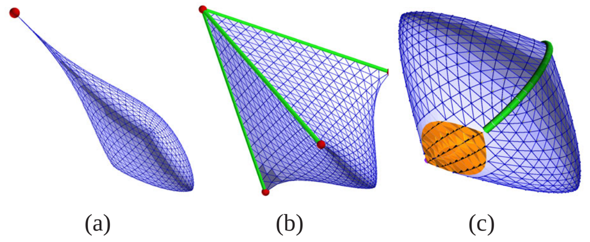
Figure 16: *Examples of singularities in manifolds. (a) A singular vertex. (b) A corner and creases. (c) A 2-manifold embedded in the 3-manifold.*

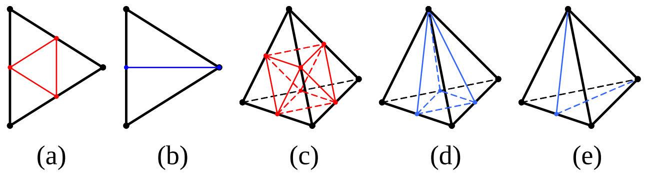
Figure 17: *Local refinement rules. (a) Red Rule and (b) Green Rule for local triangulation. (c) Red Rule, (d) Green-III Rule, and (e) Green-I Rule for local tetrahedralization.*

starting from it. The case shown in Figure 16(c) is more subtle. The user has introduced a 2-manifold singular region in the middle of the 3-manifold. As a result, the 3-manifold is split into two parts along with the singular surface. Both parts are smooth inside and outside, but the intersection is only smooth along with the tangent direction of the singularity. These types of singularities are especially useful if we want to design or fit objects with heterogeneous material. For instance, we can model a geological image containing streams and mineral veins (1- and 2-singularities) with ease.

4.2. Local Adaptive Refinement

During the process of modeling an object represented by our framework, a situation can occur, that requires finer simplices than originally given. For instance, we may want to generate very fine details on a certain region of the manifold that is defined over one simplex originally. Since the subdivision rules generate a C^1 smooth box spline on a single simplex, it is not possible to achieve high-level of detail without splitting the simplex itself. One obvious solution is a global refinement of the entire complex. This surely would work, but at the expense of the size of the complex and the memory consumption. If we simply split a single simplex, the integrity of the complex will be broken, since the neighboring simplices become non-simplicial by the introduction of cracks, or T-junctions. We follow typical Red-Green split rules to avoid the situation (See Figure 17). For the 1-simplex case, no special rule is needed. For the 2-simplex case, only the 1-ring of the adjacent simplices are affected by Green rule (Figure 17(b)). For 3-simplices, the 1-ring of the adjacent simplices are split by Green-III rule (Figure 17(d)), while the 2-ring of the neighboring simplices and the edge-sharing simplices are modified by Green-I rule (Figure 17(e)). For an octahedral cell, we simply split it into four tetrahedra, without effecting the neighbors (See Figure 18(a) and (b)). Then we can apply Red-Green rules as usual.

5. Implementation

In this section, we discuss detailed issues related to the implementation of the framework and some of results that are from our experimental design system.

5.1. Input Data

As an input, the framework takes a combination of the vertex set $\mathcal{V}$, the complex $\mathcal{C}$, and the singular subcomplex $\mathcal{S}$. However, since subsimplices can be induced from maximal simplices, we do not need all the simplices in $\mathcal{C}$. So, in the implementation, we only take the following data as an input.

- Set of vertices: $\mathcal{V} = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^3\}$.
- Set of maximal simplices:
  $\max(\mathcal{C}) = \{S \in \mathcal{C} \mid S: \textit{maximal}\}$.
- Set of maximal singular simplices:
  $\max(\mathcal{S}) = \{T \in \mathcal{S} \mid T: \textit{maximal}\}$.
- Set of combined maximal:
  $\mathcal{M} = \max(\mathcal{C}) \cup \max(\mathcal{S})$.

These are the minimum data that are required to reconstruct the complex and the other information. Additional input can include user-specific preferences for each non-manifold cases. Since we heavily rely on set operations on the complex, an efficient data structure is necessary.

5.2. Complex Construction

We reconstruct the complex $\mathcal{C}$, the decomposition $\mathcal{C}_k$, the boundary $\partial\mathcal{C}$, and the type 2 non-manifold simplices according to the following process:

- Initialize $\mathcal{C}$ as empty
- For each $k = 0,1,2,3$:

```
 1. For each $k$-simplex $S \in \mathcal{M}$:
 2.     Put $S$ in $\mathcal{C}_k$.
 3.     For each $l$-subsimplex $T \subset S$ with $l < k$.
 4.         Put $T$ in $\mathcal{C}_k$.
 5.         Construct $\rho(S)$ if $l = 0$, or 1.
 6. For each new $(k-1)$-subsimplex (face) $T$:
 7.     If ( $T$ belongs to only one $k$-simplex),
 8.         Tag $T$ as boundary.
 9.     Else if ( $T$ belongs to more than two $k$-simplex),
10.         Tag $T$ as non-manifold type 2.
```

Once the complex is constructed, we figure out the other types of non-manifold simplices. This has to be done after the construction, because we need the boundary and the decomposition information:

- For each $k = 0,1$:

```
1. For each $k$-simplex $S \in \mathcal{C}$:
2.     If ($S \in \mathcal{C}_k$ and $S \in \mathcal{C}_l$ and $k \neq l$),
3.         Tag $T$ as non-manifold type 1.
4.     Else if (type-three-test($S$)==true)
5.         Tag $T$ as non-manifold type 3.
```

Note that, the type 3 test is more complex than the others. It requires the computation of the adjacency graph of $\rho(S)$ for each 0- or 1-simplex S, and the component test:

- type-three-test(S) is true if and only if:

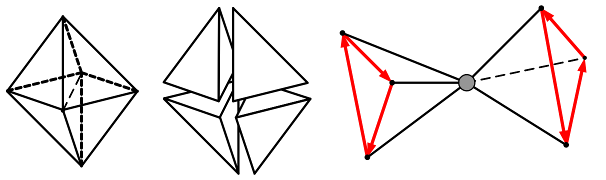

Figure 18: *(a) and (b) show a 4 split of an octahedron. (c) The 1-ring neighborhood of the type 3 non-manifold vertex. It contains 2 components.*

1. S has been tagged as boundary.
2. The adjacency graph of $\rho(S)$ contains more than one component.

Figure 18(c) shows how this process works. Once the steps are complete, we are ready to choose the appropriate subdivision rules for each vertex and edge. Note that the subsimplices induced from maximal simplices are required only for the neighborhood test and the boundary/manifold test. It can be safely removed from the memory once every step is done.

5.3. Subdivision Process

We construct the subdivision matrix and the 1-ring neighbors for each vertex and edge using the information gathered in the previous steps. Additional user input is considered to treat the non-manifold region. Then, we output $\mathcal{V}'$ and $\mathcal{C}'$ as the next level of the vertices and the complex:

- For each vertex $\mathbf{x}$ in $\mathcal{C}$:

```
1. Filter $\rho(\mathbf{x})$ to contain only the same type of vertices.
2. Choose the subdivision matrix $\mathbf{S_x}$.
3. Compute the vertex point $\mathbf{v}_{new}$ by filtered $\rho(S)$ and $\mathbf{S_x}$
4. Associate $\mathbf{v}_{new}$ with $\mathbf{x}$.
```

We follow the exactly same steps for each edge to obtain new edge points. Once the new vertex and edge points have been computed, we split each simplex:

- For each $k = 0,1,2,3$:

```
1. For each $k$-simplex $S \in \mathcal{C}$:
2.     If ($k == 0$ or 1),
3.         Put $\mathbf{v}_{new}$ or $\mathbf{e}_{new}$ associated with $S$ in $\mathcal{V}'$.
4.     Else
5.         If $S$ is an octahedron cell,
6.             Compute the cell point $\mathbf{c}_{new}$.
7.             Put $\mathbf{c}_{new}$ in $\mathcal{V}'$.
8.         Split $S$ by $\mathbf{v}_{new}$, $\mathbf{e}_{new}$ and $\mathbf{c}_{new}$if required.
9.         Put the split simplices in $\mathcal{C}'$.
```

Finally, we obtain the finer complex $\mathcal{C}'$ with the new vertices $\mathcal{V}'$. We may continue the steps from Section 5.2 to achieve more subdivision level.

5.4. Results

We have implemented a basic design system based on our framework. We present a few examples from the results of our system.

Figures 19(a-c) show a simple screw model by less than 20 control points. Four blades consist of surfaces, where the core is a solid object. The cross-section shows the inner structure of the core. In Figures 19(d-f), we use a simple spiral equation to generate the solid spring part. The valve part comprises a solid cap and a cylinder which is a surface model. All parts are represented within a single complex mesh and the non-manifold parts are smoothly blended. In Figures 19(g-i), we have designed a part of a ship that consists of the solid bow, a few decks and a part of the hull. Figures 20(a-c) illustrate a mechanical part with non-trivial topology. The handle is a 2-manifold surface model, whereas the other parts are all solid. We use the singularity rules to make the rounded corners, the sharp corners, the flat surfaces and the circular holes. The framework has the potential to be a great animation character modeling tool, as shown in Figures 20(d-f). Oftentimes, animation characters contain manifolds of different dimensions, and our framework can handle them easily. Finally, Figures 20(g-i) show an experiment with material properties. We assign tension values at the initial level, and the subdivision rules smoothly blend them into the structure. In all figures, we use the same color scheme to represent different manifolds. Blue color represents 2-manifolds or the boundaries of 3-manifolds, while orange color is for the insides of 3-manifolds. Each vertex and edge are colored according to the subdivision rules that are applied to them. Red vertices are singular, and green and blue vertices are the vertices where we apply the 1- and 2-simplex rules, respectively. The same color scheme is applied for the edges.

6. Conclusion and Future Work

We have presented a new framework for multi-dimensional adaptive subdivision objects based on simplicial complexes and subdivision schemes. A simplicial complex as a parametric domain provides us great flexibility for the topology of models. It can contain simplices of multiple dimensions simultaneously. Thus, it provides an excellent control mesh for the subdivision rules of different dimensionality. Querying and probing on the complex in our framework offers us information of topological structure of the resulting manifold. The subdivision rules based on the box splines are generalized and modified to generate manifolds of different dimensions in the limit. Unlike the tensor-product schemes, our scheme is well-defined over a simplicial domain. The subdivision rules naturally result in highly smooth manifolds, except for the extraordinary cases, where they converge to satisfy C^1 smoothness. The general rules and the user specific rules are selectively applied to the non-manifold region to model special shapes in practice. The boundary representation for each manifold is based on the subdivision rules of one lesser dimension. Therefore, the result is consistent throughout the framework. Singularities are defined as an embedded subcomplex of the domain, and the appropriate subdivision rules are applied only on the subcomplex, so that sharp features can be also represented as manifolds within manifolds. Furthermore, local refinement rules are also illustrated, which affords a user a mechanism for selective detail control on the objects. In the implementation, the properties of the complex domain are extensively employed to obtain various topological information. We also briefly discuss the analysis of the subdivision schemes, which is mostly based on well-established mathematical and numerical techniques.

Our new framework has great potential for the modeling of very complex, real-world objects. The subdivision rules can be used to approximate not only geometric models, but also material attributes of heterogeneous objects. In particular, if combined with a proper approximating algorithms, the framework can be applied to reconstruct and compress large heterogeneous models, like bio-medical images, or geo-scientific data. We are pursuing this and other directions such as data fitting, modeling of physical attributes, and model segmentation. In addition, although we have implemented tools for the basic modeling purposes, more practical operations would enable us to push the framework toward many practical applications in computer-aided design and manufacturing. These operations include, but are not limited to, set operations between manifolds, direct sculpting, and material painting.

Finally, the subdivision analysis that has been suggested here are only a glimpse of the full analysis in 3-D case. We intend to pursue more complete and general analysis of subdivision schemes on 3-simplices in the near future.

Acknowledgments

The authors wish to thank Kevin T. McDonnell for his positive suggestions and for proof-reading the paper. This research was supported in part by the NSF grants IIS-0082035, IIS-0097646, IIS-0326388, and ACR-0328930, and an Alfred P. Sloan Fellowship.

References

[BFK84] BOEHM W., FARIN G., KAHMANN J.: A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design 1* (1984), 1–60. 1

[BSWX02] BAJAJ C., SCHAEFER S., WARREN J., XU G.: A subdivision scheme for hexahedral meshes. *The Visual Computer 18* (2002), 343–356. 2, 7

[CC78] CATMULL E., CLARK J.: Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10* (Sept. 1978), 350–355. 2

[CMQ02] CHANG Y.-S., MCDONNELL K. T., QIN H.: A new solid subdivision scheme based on box splines. In *Proceedings of Solid Modeling 2002* (2002), pp. 226–233. 2, 4

[CMQ03] CHANG Y.-S., MCDONNELL K. T., QIN H.: An interpolatory subdivision for volumetric models over simplicial complexes. In *Proceedings of Shape Modeling International 2003* (May 2003), pp. 143–152. 2

[dBHR93] DE BOOR C., HÖLLIG K., RIEMENSCHNEIDER S.: *Box Splines*. Springer-Verlag, New York, 1993. 4, 7

[FMP03] FLORIANI L. D., MORANDO F., PUPPO E.: Representation of non-manifold objects through decomposition into nearly manifold parts. In *Proceedings of Solid Modeling 2003* (2003), pp. 304–309. 2

[FMPS02] FLORIANI L. D., MAGILO P., PUPPO E., SOBRERO D.: A multi-resolution topological representation for non-manifold meshes. In *Proceedings of Solid Modeling 2002* (2002), pp. 159–170. 2

[GP89] GREISSMAIR J., PURGATHOFER W.: Deformation of

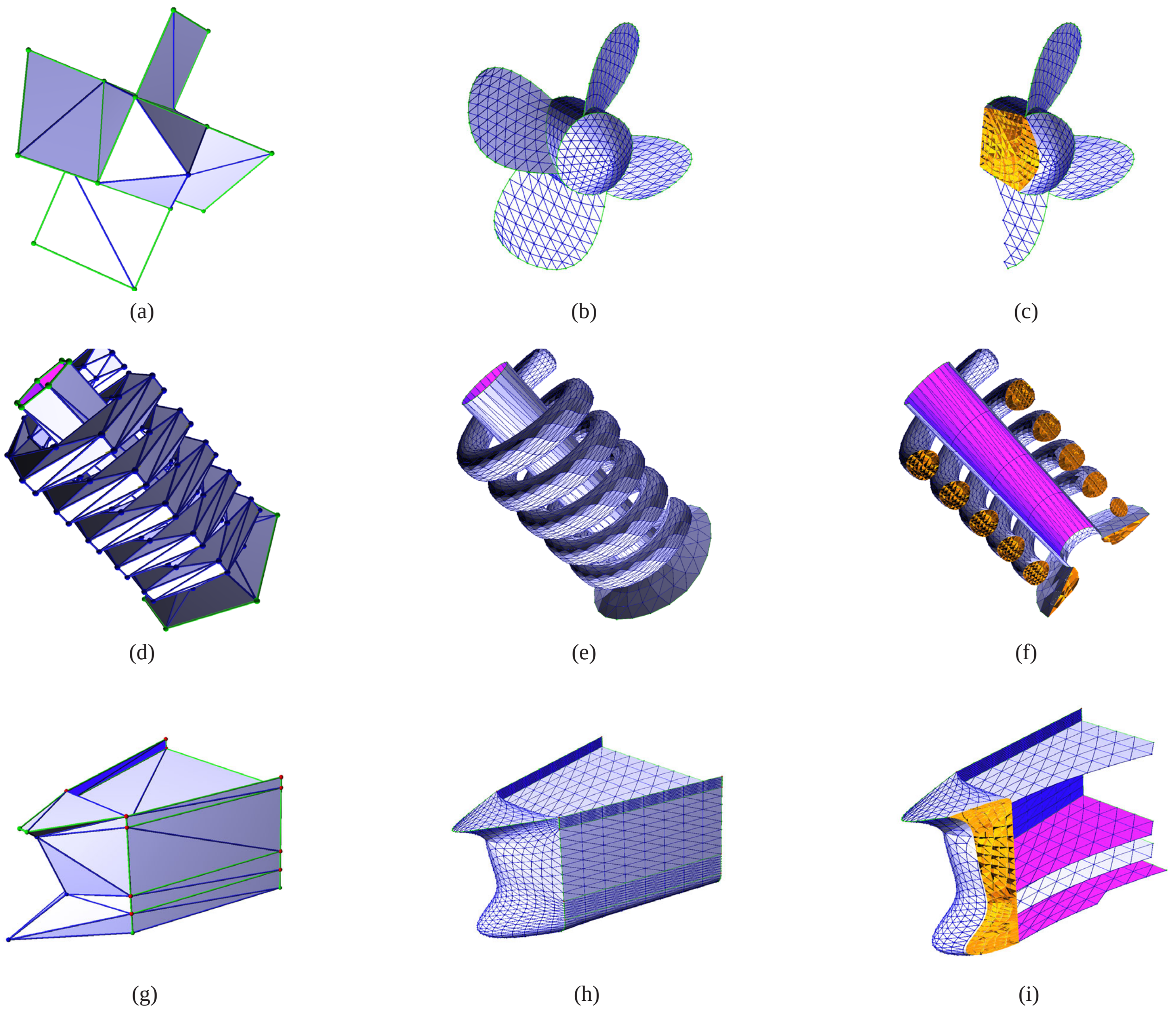

Figure 19: *Examples of non-manifold objects by our framework. (a), (d) and (g) Initial control meshes. (b), (e) and (h) After 3 levels of the subdivision. (c), (f) and (i) The cross-section of the 3-manifold regions.*

solids with trivariate B-splines. In *Computer Graphics Forum (Proceedings of Eurographics '89)* (1989), pp. 137–148. 1

[HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDONALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In *Computer Graphics (SIGGRAPH '94 Conference Proceedings)* (1994), pp. 295–302. 2, 8

[Hop96] HOPPE H.: Progressive meshes. In *Computer Graphics (SIGGRAPH '96 Conference Proceedings)* (1996), pp. 99–108. 1

[Las85] LASSER D.: Bernstein-bezier representation of volumes. *Computer Aided Geometric Design 2*, 1-3 (1985), 145–150. 1

[Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis, University of Utah, Department of Mathematics, 1987. 2, 4, 6

[MJ96] MACCRACKEN R., JOY K. I.: Free-Form deformations with lattices of arbitrary topology. In *Computer Graphics (SIGGRAPH '96 Conference Proceedings)* (1996), pp. 181–188. 2

[MP87] MICCHELLI C., PRAUTZSCH H.: Computing surfaces invariant under subdivision. *Computer Aided Geometric Design 4*, 4 (1987), 321–328. 7

[PH97] POPOVIC J., HOPPE H.: Progressive simplicial complexes. In *Computer Graphics (SIGGRAPH '97 Conference Proceedings)* (1997), pp. 217–224. 2

[Pra85] PRAUTZSCH H.: Generalized subdivision and convergence. *Computer Aided Geometric Design 2*, 1-3 (1985), 69–76. 7

[Pra98] PRAUTZSCH H.: Smoothness of subdivision surfaces

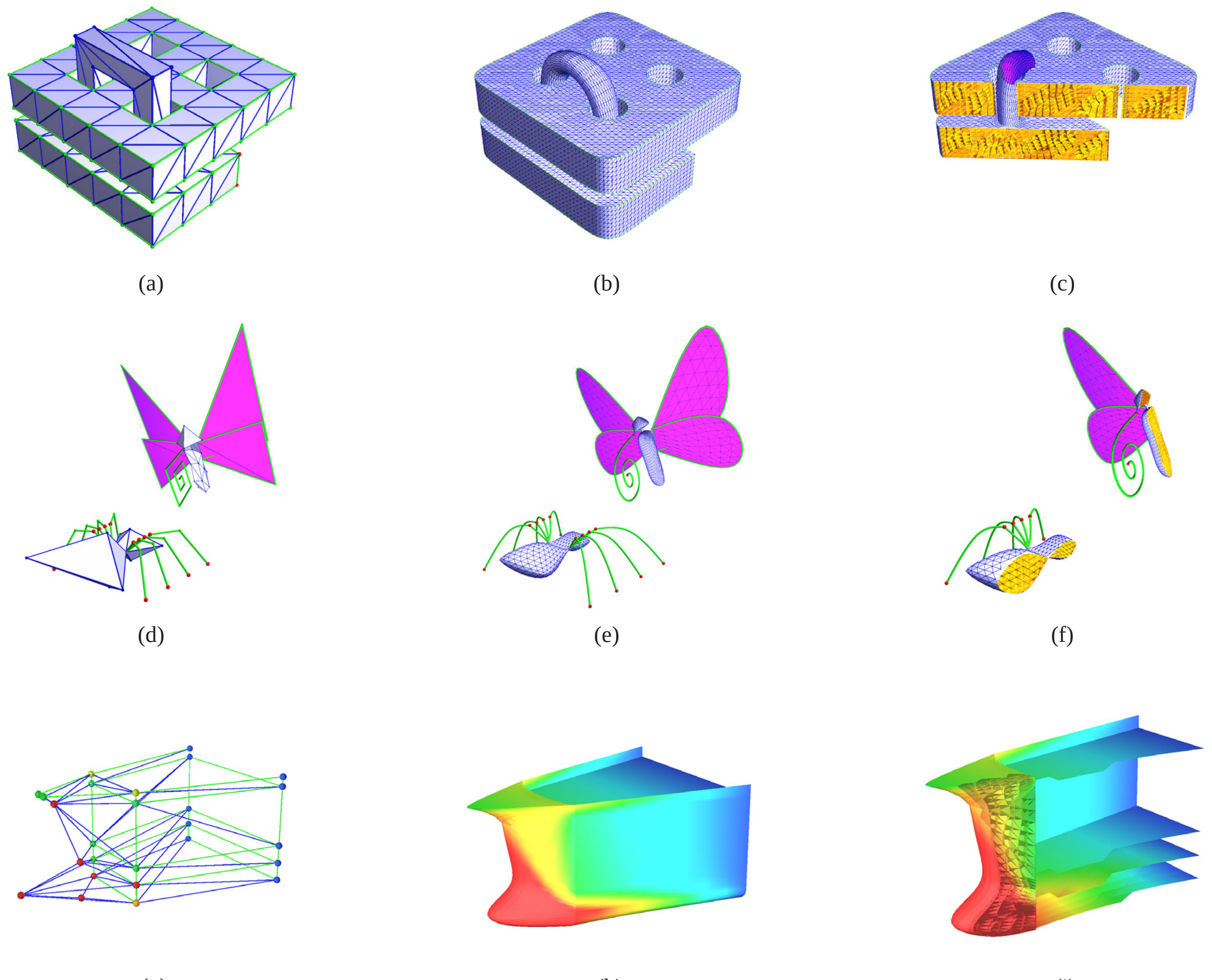
(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 20: *Examples of non-manifold objects and material property representations by our framework. (a), (d) and (g) Initial control meshes. (b), (e) and (h) After 3 levels of the subdivision. (c), (f) and (i) The cross-sections of the 3-manifold regions.*

at extraordinary points. *Advances in Computational Mathematics 9* (1998), 377–390. 7

[PS94] PASKO A. A., SAVCHENKO V. V.: Blending operations for the functionally based constructive geometry. In *CSG 94 Set-theoretic Solid Modeling: Techniques and Applications* (1994), pp. 151–161. 1

[Rei95a] REIF U.: Some new results on subdivision algorithms for meshes of arbitrary topology. *Approximation Theory VIII 2* (1995), 367–374. 7

[Rei95b] REIF U.: A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design 12* (1995), 153–174. 7

[RV82] REQUICHA A. A. G., VOELCKER H. B.: Solid modeling: a historical summary and contemporary assessment. *IEEE Computer Graphics and Applications 2* (Mar. 1982), 9–23. 1

[WMW86] WYVILL B., MCPHEETERS C., WYVILL G.: Animating soft objects. *The Visual Computer 2*, 4 (1986), 235–242. 1

[WW01] WARREN J., WEIMER H.: *Subdivision Methods for Geometric Design: A constructive Approach*. Morgan Kaufmann Publisher, 2001. 5, 6, 7

[YZ01] YING L., ZORIN D.: Nonmanifold subdivision. In *Proceedings of IEEE Visualization 2001* (2001), pp. 325–332. 2, 6

[Zor00] ZORIN D.: Smoothness of stationary subdivision on irregular meshes. *Constructive Approximation 16* (2000), 359–398. 7

[ZSS96] ZORIN D., SCHRÖDER P., SWELDENS W.: Interpolating subdivision for meshes with arbitrary topology. In *Computer Graphics (SIGGRAPH '96 Conference Proceedings)* (1996), pp. 189–192. 7