Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer-Aided Design 38 (2006) 770–785
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

A unified subdivision approach for multi-dimensional non-manifold modeling

Yu-Sung Chang[a,*], Hong Qin[b]

[a] *Wolfram Research, Inc., 100 Trade Center Drive, Champaign, IL 61820-7237, USA*
[b] *Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400, USA*

Received 12 September 2005; accepted 11 April 2006

Abstract

This paper presents a new unified subdivision scheme that is defined over a k-simplicial complex in n-D space with $k \leq 3$. 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-manifolds 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 subdivision schemes to show potential benefits of our work in industrial design, geometric processing, and other applications.

Keywords: Subdivision algorithms; Geometric and topological representations; Solid modeling; Multiresolution models; Volumetric meshes

1. Introduction

1.1. Motivation

Many industrial design projects include a wide range of shape representations in a single place. For instance, in car design, a hood of a car can be represented as thin plate, while volume representation is more appropriate for the engine parts. Such a situation leads to complicated non-manifold objects, where curves and surfaces meet together, or multiple surfaces coincide in one edge. In addition, boundary and feature representations are essential in mechanical design. Without modification, subdivision schemes tend to smooth objects, since the subdivision process is weighted averaging in essence. 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:

- Define a parametric domain that provides high flexibility in modeling and simplicity in topological inquiry.
- Represent objects with multiple dimensions in a single framework.
- Develop unified subdivision rules for arbitrary manifolds and multiple dimensions.
- Automatic treatment for non-manifold regions with minimal user intervention.
- Support the boundary and the sharp feature representation
- Level-of-detail (LOD) control.

Fig. 1 illustrates a general idea of our approach. In the next section, we discussion by reviewing the previous work that is related to the goals specified above. First, we begin our background review with brief mentioning of current existing representations in Solid Modeling. Then, we

* Corresponding author. Tel.: +1 217 398 0700; fax: +1 217 398 0747.
E-mail addresses: yusung@wolfram.com (Y.-S. Chang), qin@cs.sunysb.edu (H. Qin).

0010-4485/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2006.04.004

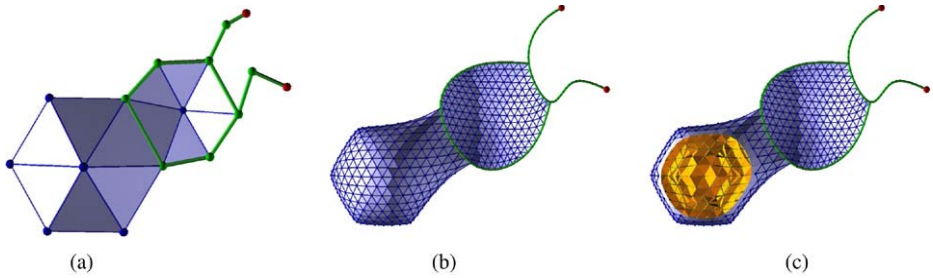

Fig. 1. A non-manifold object represented by our subdivision scheme. (a) The initial complex that consists of 1-, 2-, and 3-simplices. (b) After level 3. (c) The cross-section of the 3-manifold reveals the internal structure.

review the previous research related to parametric domains, subdivision schemes, and their expansions such as non-manifold representation and feature representation.

1.2. Background

Since Requicha and Voelcker's [26] 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. As Boehm et al. [2] surveyed, parametric curves and surfaces had been widely used especially in computer-aided design and manufacturing for a long time. Bernstein-Bézier solids [14], B-spline solids, and other tensor product based [12] approaches are typical examples of parametric representations in solid modeling. Implicit function methods, such as CSG [20] and blobby models [29], 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 though the implicit function method 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 in exchange of analytic geometric information.

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 general, the subdivision technique has the following advantages:

- Uniformity of representation
- Multiresolution analysis and levels of details
- Numerical efficiency and stability
- Arbitrary topology or genus.

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 [3] and the Multi-linear Cell Averaging scheme by Bajaj et al. [1] both utilize tensor-product cubic B-splines. In any case, their parametric domains should have the form of a tensor-product space. Some shortcomings 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 have lattice structure, which is less desirable than triangular or tetrahedral meshes when one wants to represent unstructured shapes. 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. [10,11] 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 [16]. For 3-D solid objects, MacCracken and Joy [17] 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. [1] further extended the scheme with an analysis based on numerical experiments. They are both tensor-product extensions of the cubic B-spline curves, and hence are not suitable for our purpose. Pascucci [19] suggested a special solid subdivision scheme with slow cell increase. Most recently, Schaefer et al. proposed a tetrahedral mesh based subdivision [27]. Interestingly, they use the octet-truss structure and prove the

smoothness using the joint spectral analysis recently developed by Levin et al. [15]. In addition, Chang et al. [4] 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 [5] 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.

Non-manifold subdivision

Non-manifold regions can occur through self-intersection in a single dimension. Even though subdivision approaches have a benefit of topological flexibility over other modeling representations, it is not trivial to deal with a non-manifold situation. Ying and Zorin [30] 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 subdivision scheme with a single dimension, since it involves intersections between splices of various dimensions. Not only does our approach provide a specific solution for each situation, but also we suggest a generalized rule based on a solid subdivision scheme.

Feature and detail control

Generally speaking, 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. [13] 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. [21] presented the idea of a progressive simplicial complex. In this paper, 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 and demonstrates several models generated by our framework. Finally, we discuss future work and conclude the paper in Section 6.

2. Simplicial complex domain

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 a 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.

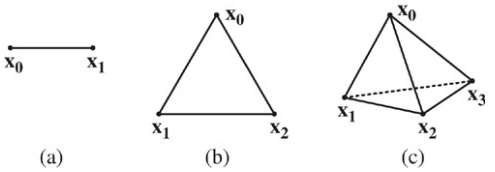

Fig. 2. Examples of simplices. (a) A 1-simplex, (b) a 2-simplex, and (c) a 3-simplex.

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, the subdivision rules are not only homeomorphic, but their limits also satisfy the higher level of smoothness on the 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. Set definitions

Our domain of choice is a simplicial complex in $\mathbb{R}^n$ (see Fig. 2). A k-simplex S can be defined as a set in $\mathbb{R}^n$,

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

where

$$c_i \geq 0, \quad \sum_{i=1}^{k} c_i = 1, \quad \mathbf{x}_i \in \mathbb{R}^n. \tag{2}$$

Since S can be uniquely determined by $k+1$ points $\mathbf{x}_0$, $\mathbf{x}_1, \ldots, \mathbf{x}_k$, and is independent of their ordering, we simply use a set notation $S := \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$. In this paper, we limit k to be less than or equal to three. Also, we consider each simplex as a closed set. Note that any subset of S also represents 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 $\dim(S)$.

A simplicial complex, or a complex, $\mathcal{C}$ is a finite collection of simplex representing sets S where every non-empty subset of S is also an element of $\mathcal{C}$. Its geometric interpretation is as follows: (1) the simplices represented by the subsets of each S 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.

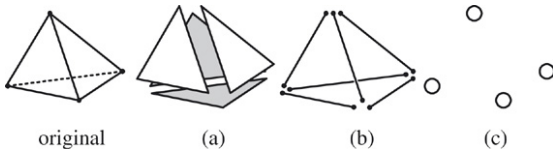

Fig. 3. The subsimplices of a 3-simplex. (a) The 2-subsimplices, (b) the 1-subsimplices, and (c) the 0-subsimplices.

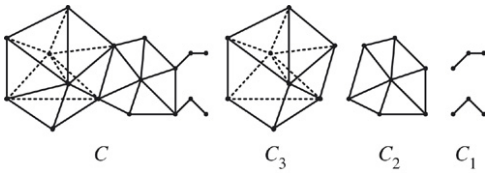

Fig. 4. Complex decomposition. A complex $\mathcal{C}$ can be decomposed into $\mathcal{C}_k$s with $k = 1, 2, 3$.

In the complex $\mathcal{C}$, we call a simplex a *subsimplex* if it is a subset of other members of the complex (see Fig. 3). Likewise, it is called a *proper subsimplex* if it is a proper subset of a simplex. In addition, a simplex is called a *maximal* simplex if it is not a subsimplex of any other simplices in $\mathcal{C}$.

In summary, the domain space of our framework can be expressed as a pair of the following sets ($|\cdot|$ represents the number of elements in a set):

- Set of vertices

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

- A simplicial complex:

$$\mathcal{C} = \{S \subset \mathcal{V} \mid S \neq \emptyset, |S| \leq n+1\}, \tag{4}$$

with the following property:

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

2.2. Complex decomposition

A complex $\mathcal{C}$ can contain simplices of different dimensions (see Fig. 4). 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 simplices always have the dimension k. 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

$$\mathcal{C} = \bigcup_{k=1,\ldots,m} \mathcal{C}_k, \tag{6}$$

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

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

We should mention that each $\mathcal{C}_k$ can contain several components, or maximal subcomplexes. In our approach, this information will not be utilized, even though it can be useful for other applications. In Section 3, we define k-manifolds (with boundary) over the k-subcomplex using appropriate subdivision rules. However, the $\mathcal{C}_k$ 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 simplex

A face of a k-simplex S is simply defined as a $(k-1)$-subsimplex of S. Even for $k \neq 3$ cases, we still opt to use the word "face" for any immediate subsimplices of k-simplex, due to its geometric implication. A boundary of a complex can be defined as follows:

- 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 $\partial\mathcal{C}$.

2.4. Non-manifold simplex

If our domain consists of a single k-simplex, it is trivial to establish a manifold map from the simplex to a k-manifold. However, it is not always possible to define a manifold map over a complex. For instance, if the domain consists of a 2-simplex and a 3-simplex joined by an edge, it is not possible to define either a single 1- or 2-dimensional Euclidean map across the edge. Also, if three or more 2-splices share a single edge in general position, we cannot find any single Euclidean map that can be well-defined across the edge. These cases occur only on the intersections of the simplices that comprise the domain. We call a simplex a *non-manifold* simplex, if we cannot define a Euclidean map on the simplex. The following definition covers all the possibilities of non-manifold simplices:

- Non-manifold simplex: A k-simplex $S \in \mathcal{C}$ is a non-manifold simplex, if it satisfies any of the following properties.
  (1) $S \in \mathcal{C}_k \cap \mathcal{C}_l$ where $k \neq l$ (see Fig. 5 (a)).
  (2) $S \in \mathcal{C}_k$ exclusively, $\dim(S) = k-1$ and $S = S_1 \cap S_2 \cap S_3$ for some distinct k-simplices $S_1, S_2, S_3 \in \mathcal{C}_k$ (see Fig. 5 (b)).
  (3) $S \in \mathcal{C}_k$ exclusively, $\dim(S) < k-1$, $S = S_1 \cap S_2$ for some maximal k-simplices $S_1, S_2 \in \mathcal{C}_k$, $S_1 \neq S_2$ and S is not a proper subsimplex of any non-manifold simplex (see Fig. 5 (c)).
  (4) $S \in \mathcal{C}_k$ exclusively, $\dim(S) < k-1$ and S is a subsimplex of a non-manifold simplex.

Any given non-manifold simplex should satisfy one of the properties, but not both. We call the first three cases type 1, type 2 and type 3 non-manifold simplices, respectively. Type

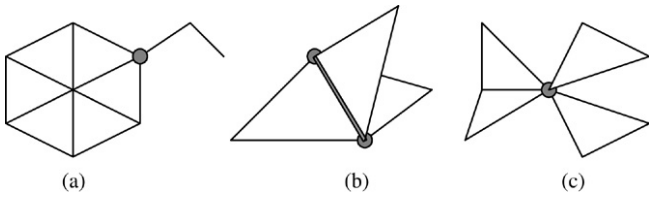

Fig. 5. Examples of complexes containing non-manifold simplices. (a) Type 1, (b) type 2, and (c) type 3. The vertices or edges in gray are the non-manifold simplices. The vertices of the gray edge in (b) are categorized as type 4.

4 explains the subsimplex cases of non-manifold simplices, which are not non-manifold simplices by themselves. We employ various strategies to tackle the non-manifold cases. Generally, non-manifold simplices create ill-posed problems. To be exact, 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. The fourth case can be dealt with the same as the solutions for the other three cases. Fig. 5 shows examples of these cases.

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 $\mathcal{C}$. As the function $N(\mathbf{x})$, we choose the box spline whose support lies in the 1 vertex neighbors of simplices. Otherwise specified, we use the term *1-ring* of a vertex $\mathbf{v}$ to represent the adjacent vertices of $\mathbf{v}$ where the complex which only contains these neighbor vertices does not have the boundary of its own. 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 [16] for surfaces. For 3-D, we use the box spline solid that has been employed in our previous work [4]. Non tensor-product box splines are particularly 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; (3) The choice of domain is more flexible.

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 $\delta_i \in \mathbb{R}^n$ 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:

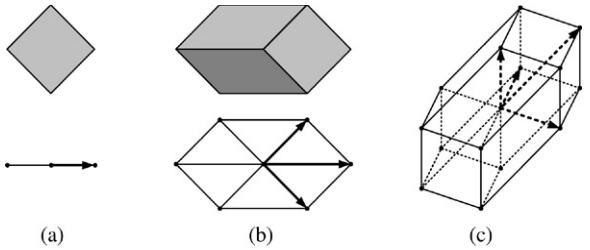

Fig. 6. The domain support for the box splines. The upper images are the unit cubes whose projections are taken. The thick arrows are the direction vectors. For (c), we only display the support, since it is hard to visualize a 4-hypercube.

(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 Fig. 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 Fig. 6(c). Unfortunately, it is not embedded in the 1-ring of 3-simplices, or tetrahedra. However, by adding few more edges and faces, we can turn it into a simplicial complex.

Generally, the box splines satisfy the following properties, as proven in [7]:

(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 $k+2$.

3.2. Subdivision meshes

The box splines can be expressed as a sum of the box splines with the half-sized supports (see Fig. 7). Using this property, we can find out the rules for the subdivision scheme for regular cases. 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 Fig. 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

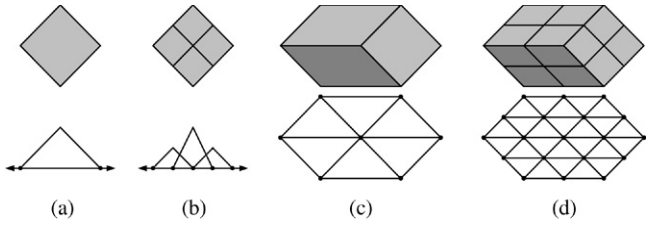

Fig. 7. Subdivision of the box splines. (a) and (c) show the cubes and their projected domains. (b) and (d) show the trivial subdivision of the cubes and the domains. The linear case spline is drawn in (a) and (b).

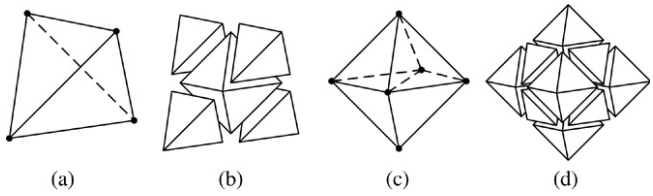

Fig. 8. Split a tetrahedron and an octahedron.

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 $\mathbb{R}^2$ (see [6,32]). We resolve this problem with the regular case by the following approaches:

(1) The boundary of the projection of a 4-hypercube on $\mathbb{R}^3$ (see Fig. 6(c)) is a rhombic dodecahedron. It is well-known that this polytope can tile the space.
(2) By introducing a few additional edges, we can decompose the dodecahedron into several tetrahedra.
(3) A single tetrahedron can be split into four congruent tetrahedra and one octahedron, as shown in Fig. 8(b). Also, an octahedron can be split into eight tetrahedra and six congruent octahedra (see Fig. 8(d)).
(4) If we keep continuing this process, then we get a semi-regular space-filling structure called an *octet-truss* (see Fig. 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 with the subdivision of the 3-simplex domain.
(5) We store one diagonal inside an octahedron, as shown in Fig. 8(c), to keep track of the adjacency of each vertex. In fact, each octahedron can be considered as a family of four tetrahedra.

Please remind that this approach is only for the regular rules. Extraordinary cases will be discussed and analyzed in the latter part of the paper.

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. We follow the generating function method, first explored by Dyn and Micchelli [9]. It is known that the

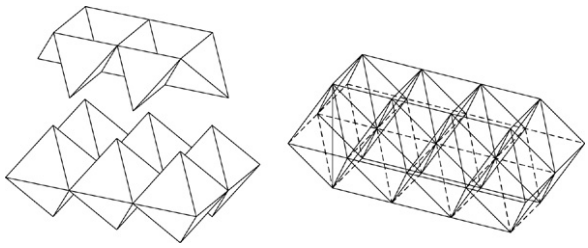

Fig. 9. Octet-truss.

coefficients of the generating functions can provide us the coefficients for the subdivision rules, as proven in [28]. 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 $\mathbf{z}^{\delta_i}$ 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 $\mathbf{x}_i$):

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

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

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

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

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

Figs. 10 and 11 summarizes the regular configuration and the rules. Here, we use more conventional names for 0-, 1-, 2-, and 3-simplices, namely, vertices, edges, faces, 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^2 smoothness as mentioned above.

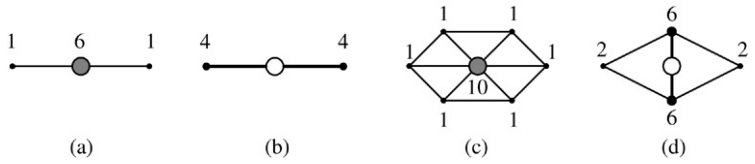

Fig. 10. Regular subdivision rules. (a) The 1-simplex rules. (b) The 2-simplex rules.

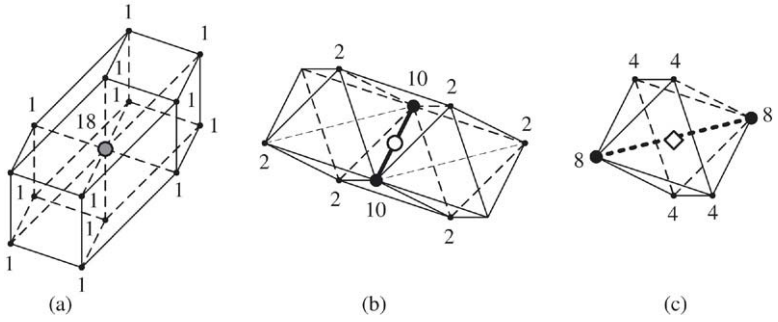

Fig. 11. Regular 3-simplex subdivision rules. (a) Vertex point, (b) edge point, and (c) cell point rules.

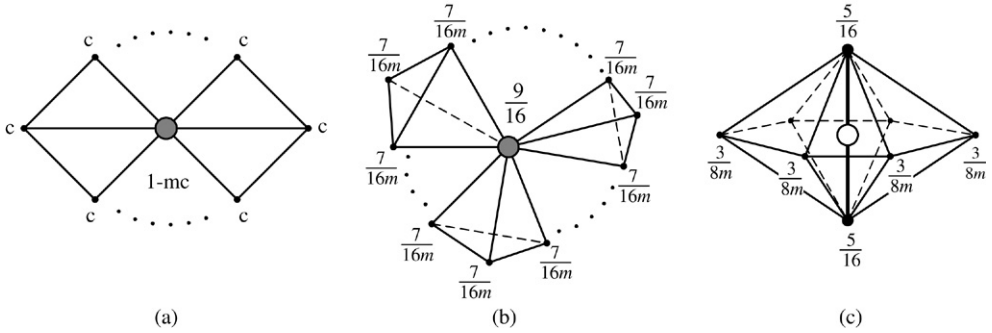

Fig. 12. Modified k-simplex subdivision rules.

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 become isolated during 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 [16] 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. (see [28, Section 7.3.2]):

- Modified 2-simplex subdivision rules:

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

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

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

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

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

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

Fig. 12 illustrates the modified rules in general configurations.

3.5. Boundary and non-manifold rules

The rules introduced in the previous section cannot be applied to some special cases, such as boundaries, non-manifold regions and singularities. We will discuss the

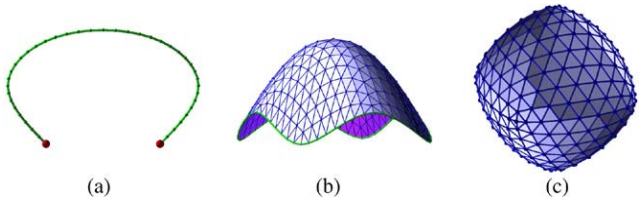

Fig. 13. Examples of manifolds with boundary. (a) A 1-manifold with boundary. (b) A 2-manifold with boundary. (c) A 3-manifold with boundary.

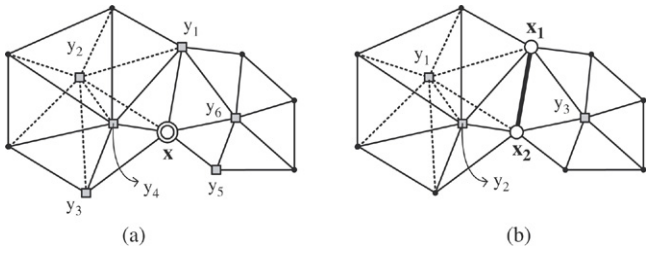

Fig. 14. The 1-ring neighbors with the relieved topology condition. (a) shows an example of the 1-ring of the vertex **x** for type 3 non-manifold vertices. Under the relieved condition, we choose all its adjacent vertices. (b) shows an example of the 1-ring neighbors of the edge $[\mathbf{x_1}, \mathbf{x_2}]$.

singularities in Section 4.1. 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. Fig. 13 demonstrates examples of such boundary cases. Non-manifold regions require special rules. 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 and Zorin [30] proposed detailed approaches to overcome non-manifold topology with subdivision surfaces. They involve 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 solution is much simpler, as described below. For each case, we can apply either specific rules (N-1 and N-2) or general rules (G-1 and G-2):

- The following three rules are specific for type 1 and type 2 cases.
  - **Rule N-1.** Type 1 is a region where the manifolds with different dimensions meet. In this case, we can follow the subdivision rules for a single simplex of a user's choice. The region is only explained by the subdivision rules of the chosen dimension, and other cells with different dimensions only maintain the connectivity.
  - **Rule N-2.** Type 2 is a region where a multiple manifold of a single dimension intersects by their faces. This region can be considered as a self-intersection. Our suggested solution is to choose one pair of the simplices on which we apply the subdivision rule.
- Type 3 is a region where multiple manifolds of a single dimension intersect, but they do not share faces. In this case, we found that the general rules described below yield the best results.
- 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 relieved topology condition.

By a subdivision rule with the *relieved topology condition*, we mean that the rule only considers the connectivity between each vertex when acquiring the 1-ring of neighbors, even if it does not satisfy the condition for a 1-ring that is defined in the previous section. Fig. 14 shows examples of neighbor choices by the relieved topology condition. Since the intersection between simplices with different dimensions always occurs on the boundary complex $\partial\mathcal{C}$, we only choose the neighbors in $\partial\mathcal{C}$. Figs. 15 and 16 illustrate examples of non-manifold cases. In Fig. 15(a), a 1-manifold (Part A) intersects with a 3-manifold (Part B). Therefore, it forms a type 1 case. Also, a type 3 non-manifold case is shown between Parts B and C. For these particular cases, a user has not specified the rule. Thus, the system follows rule G-2, which results in the smooth blending of the manifolds. Fig. 15(b) and (c) show typical type 2 cases. In Fig. 15(b), four 2-manifolds intersect together at a line (Line $\overline{AA'}$). A surface self-intersects in Fig. 15(c). For both cases, we use rule G-2 to blend non-manifold parts into the bodies. Fig. 16 shows the effects of the different rules. In Fig. 16(b), the user selects vertex A 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

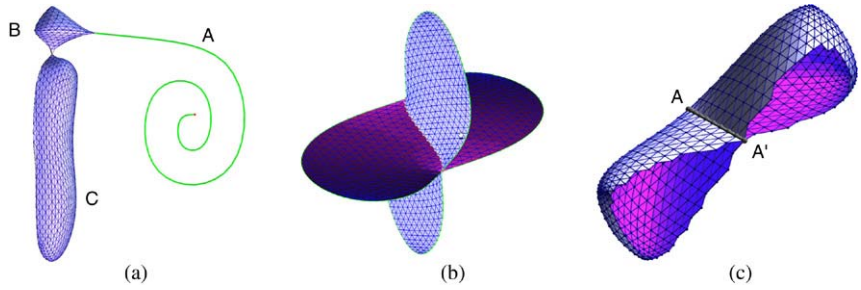

Fig. 15. Examples of non-manifold cases. (a) A type 1 case by the 1- and 3-manifold intersection. Also, a type 3 case is shown between two solid parts. Rule G-2 is applied in both the cases. (b) A type 2 case by 2-manifold intersection. (c) The cross-section of another type 2 case.

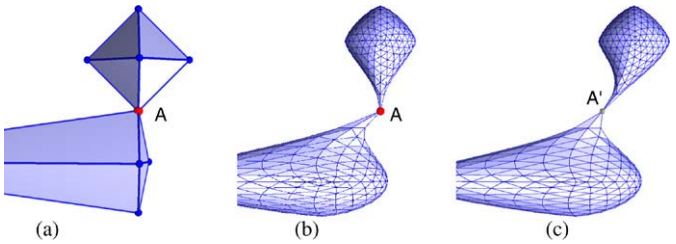

Fig. 16. Type 3 non-manifold rules. (a) The initial complex. (b) The subdivision by rule G-1. (c) The subdivision by rule G-2. Vertex A preserves its position in (b), while it is blended in (c).

during the subdivision. However, in Fig. 16(c), we follow rule G-2. As a result, the vertex has been moved (A') 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. Fig. 17 lists all the solutions provided by our subdivision scheme for a single configuration. Overall, rule G-2 provides the most visually pleasing results. 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. Subdivision 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 [7,28]. 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 developed by Doo and Sabin [8] and improved by many researchers. For instance, Micchelli [18], Prautzsch [22,23], Reif [25,24], and more recently, Zorin [33,31] investigated the sufficient and necessary conditions of convergence and the C^1 smoothness. Since our 1- and 2-simplex rules exactly follow the rules that already have been analyzed by other research, we focus ourselves on the 3-simplex, i.e., the solid subdivision rules.

Since the subdivision process is a linear combination, in essence, we can represent the rules locally by the subdivision matrix $\mathbf{S}$,

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

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 the λ_i are the (left) eigenvalues of $\mathbf{S}$ 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 $\mathbf{S}$,

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

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:

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

As shown in Fig. 12(a), the matrix $\mathbf{S}$ 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 $\mathbf{S}$ 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

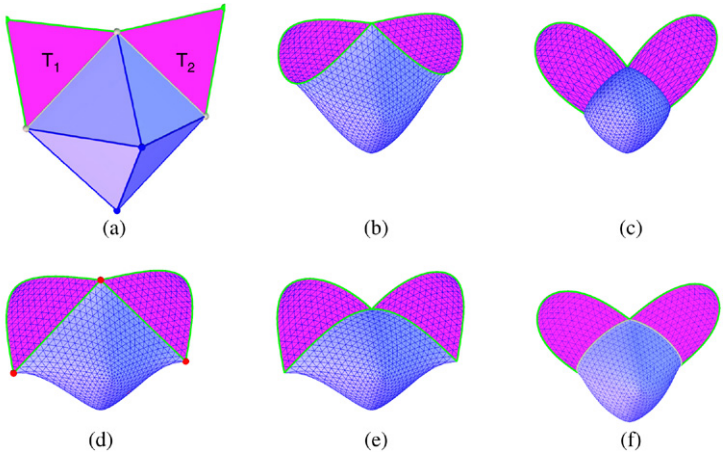

Fig. 17. Comparison between the non-manifold rules. (a) The initial control points. The complex consists of one 3-simplex (octahedron) and two 2-simplices (triangles T_1 and T_2). The intersection between the 3-simplex and 2-simplices form type 1 cases. (b) Rule N-1 is applied. In this case, we consider the intersection as a part of the boundaries of the 2-simplices (triangles). (c) Rule N-1 is applied. But, instead of using 2-simplex boundary rules, we utilize the intersection as a part of the boundary of the 3-simplex. As a result, the boundary of the 3-simplex region does not change at all. (d) We apply rule G-1 with the vertices as 1-singularities. (e) We apply rule G-1 with the edges as 2-singularities. The intersection creates a 1-singular curve on the surface of the 3-simplex boundary. (f) Finally, rule G-2 is applied to the intersection. No information is specified by the user. Only connectivity and the 3-simplex subdivision rule is used. In the end, the intersection is smoothly blended.

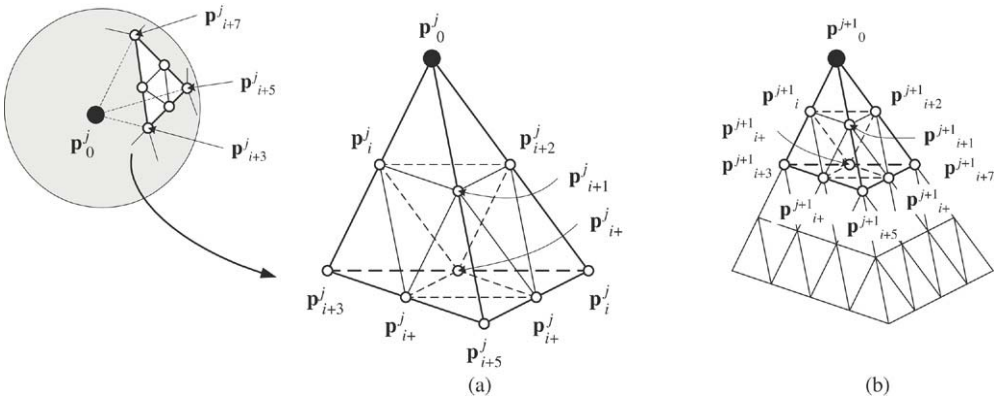

Fig. 18. The invariant neighborhood of an extraordinary vertex and their indices.

analysis on the characteristic map suggested by Reif [25] can guarantee the C^1 smoothness around the vertex. However, the subdivision matrix for the 3-simplex rules 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. [1] suggested the condition for C^1 smoothness of the three dimensional case as:

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

through their empirical analysis. We begin our analysis with computing the subdivision matrices for 3-simplex cases.

3.7. Subdivision matrix

We first examine the case of extraordinary vertices. This case involves a vertex with k vertices adjacent to it. As shown in Fig. 18, we can establish a correspondence between the k adjacent vertices and k vertices on the sphere centered by the extraordinary vertex $\mathbf{p}^j{}_0$. By considering different triangulations of the k vertices, we can understand the different configurations of the extraordinary vertex subdivision matrix. Each triangle is associated with the tetrahedral area that is surrounding the extraordinary vertex. Because we need 2-ring vertex neighbors to acquire the invariant system, we subdivide each tetrahedron once, as illustrated in Fig. 18. Using the Poincaré formula and the relation between triangular faces and edges:

$$v - e + f = 2, \tag{25}$$

$$2e = 3f, \tag{26}$$

we can deduce that the number of such tetrahedral areas surrounding the vertex is $f = 2(k-2)$. In addition, the 1-ring vertex neighbor contains k vertices and each subdivided triangular face on the 2-ring vertex neighbor contains six vertices, three of which are shared by each edge. Therefore, the actual number N of vertices including the extraordinary vertex to form the invariant system is:

$$\begin{aligned} N &= 1 + k + 6f - 3e + k \\ &= 1 + k + 6(2k-4) - 3(3k-6) + k = 5k - 5. \end{aligned} \tag{27}$$

Hence, we can conclude that the size of the subdivision matrix for each extraordinary vertex with the valence k is $N \times N$ where $N = 5k - 5$. With a proper reordering of the indexes of the vertices, the matrix $\mathbf{S}_v$ can be written as:

$$\mathbf{S}_v = \begin{pmatrix} \mathbf{M} & \mathbf{O} \\ \mathbf{A} & \mathbf{B} \end{pmatrix}, \tag{28}$$

where $\mathbf{M}$ is a $(k+1) \times (k+1)$ matrix associated with the extraordinary vertex and k adjacent vertices and $\mathbf{O}$ is the zero matrix with the size of $(4k-6) \times (k+1)$. It is important to know that the dominant and the subdominant eigenvalues of the $\mathbf{S}_v$, especially the first five largest eigenvalues, are identical to those of the submatrix $\mathbf{M}$. Since the matrix $\mathbf{M}$ can be easily acquired by the k 1-ring vertex neighbors of the vertex $\mathbf{p}^j{}_0$, we can reduce the amounts of the computations during the analysis process significantly. It is worth mentioning that, unlike the surface cases, there exist several different configurations of neighboring vertices for each valence k. Since each configuration yields a unique subdivision matrix, it is difficult to compute the eigensystem systematically.

The extraordinary edge with the valence k is surrounded by k tetrahedra sharing the edge $e = [\mathbf{p}^j{}_0, \mathbf{p}^j{}_2]$, as shown in Fig. 19. Again, we subdivide each tetrahedron once to make the neighbor invariant. It is easy to deduce that the size of the subdivision matrix $\mathbf{S}_e$ is $(4k+3) \times (4k+3)$. Similar to the extraordinary vertex subdivision matrix, the matrix $\mathbf{S}_e$ can be described as:

$$\mathbf{S}_e = \begin{pmatrix} \mathbf{L} & \mathbf{O} \\ \mathbf{P} & \mathbf{Q} \end{pmatrix}, \tag{29}$$

with the proper index reordering. In the edge case, $\mathbf{L}$ is a $(2k+3) \times (2k+3)$ matrix. It consists of the subdivision coefficients of the 1-ring neighbors of the extraordinary edge. Once more, the dominant and subdominant eigenvalues of the subdivision matrix $\mathbf{S}_e$ can be acquired from the submatrix $\mathbf{L}$.

3.8. Eigenvalues and characteristic maps

Once we acquire the subdivision matrix of an individual case, we numerically compute the eigenvalues to confirm the satisfaction of the condition (24). In Tables 1 and 2, we list a

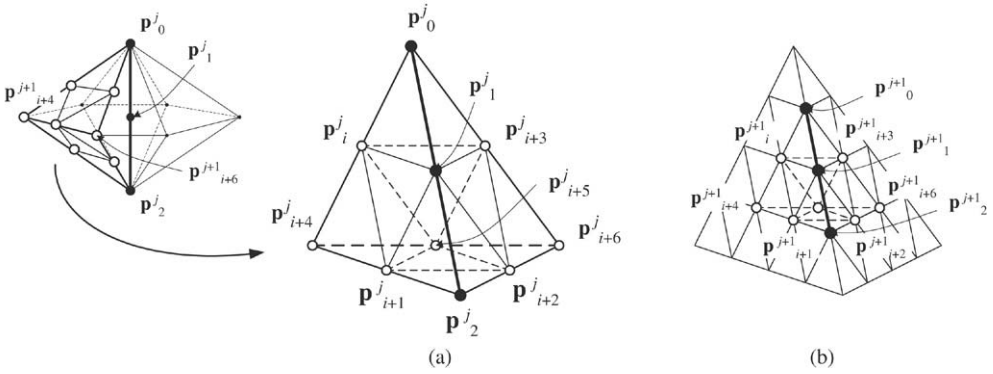

Fig. 19. The invariant neighborhood of an extraordinary edge and their indices.

Table 1
Eigenvalues for a selection of the extraordinary vertex cases

Valence	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5
5	1.0	0.3125	0.292083	0.15	0.125	0.125
6	1.0	0.312499	0.25	0.25	0.25	0.15
7	1.0	0.327254	0.327254	0.3125	0.275888	0.15
8	1.0	0.480205	0.3125	0.3125	0.249998	0.2375
9	1.0	0.405872	0.405872	0.3125	0.26545	0.19437
10	1.0	0.477404	0.418566	0.418566	0.2375	0.206434
11	1.0	0.441511	0.441511	0.3125	0.293412	0.293412
12	1.0	0.480205	0.480205	0.480205	0.250002	0.2375
13	1.0	0.460313	0.460313	0.353854	0.353854	0.3125
14	1.0	0.577132	0.449431	0.449431	0.34832	0.3125
15	1.0	0.471364	0.471364	0.392016	0.392016	0.3125
16	1.0	0.541169	0.541169	0.480204	0.372645	0.372645
17	1.0	0.571212	0.511703	0.511703	0.371472	0.358853
18	1.0	0.623289	0.463128	0.463128	0.457191	0.374739
20	1.0	0.571212	0.549072	0.549072	0.3875	0.3875
22	1.0	0.616629	0.525774	0.525774	0.4625	0.427853

Table 2
Eigenvalues for a selection of the extraordinary edge cases

Valence	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5
4	1.0	0.477404	0.418566	0.418566	0.2375	0.206434
5	1.0	0.480205	0.480205	0.480205	0.25	0.2375
6	1.0	0.517404	0.517404	0.480205	0.3125	0.3125
7	1.0	0.541169	0.541169	0.480204	0.372645	0.372645
8	1.0	0.557148	0.557148	0.480205	0.418566	0.418566
9	1.0	0.568361	0.568361	0.480206	0.453454	0.453454

selection of eigenvalues that we examined. They all satisfy the suggested eigenvalue condition.

In addition to the eigenvalue condition, we have performed the characteristic map analysis for extraordinary vertex cases. For the 2-simplex vertex cases, it is possible to do the analysis symbolically due to their symmetry. However, as mentioned earlier, the 3-simplex vertex cases do not have such symmetry. Therefore, we rely on the numerical results. We choose an extraordinary vertex or edge case. Then, we compute the eigenvectors $\mathbf{v}_i$ from the subdivision matrix $\mathbf{S}$. Afterwards, we follow the steps explained by Reif [25]. Figs. 20 and 21 show the control nets for selected extraordinary cases. Nonetheless, our experiments strongly suggest that there are no visible degenerations of the 3-manifold even after the very large number of subdivision processes.

For non-manifold cases, all the non-manifold rules, except the rule G-2, intentionally introduce singularities. For the case of rules N-1 and N-2, the limit object is smooth only along the chosen direction, whose smoothness can be easily explained by the analysis of a single dimensional subdivision rule. Across all the other directions, it is clear that it is continuous, since the rules form a convergent and surjective mapping. However, the limit region is not smooth along those directions, which is intentional. For rule G-2, we assume that all the vertices are spatially embedded in a 3-manifold, therefore the same analysis can be applied as the 3-simplex case. In this case, our limit object is C^1 smooth along all the directions.

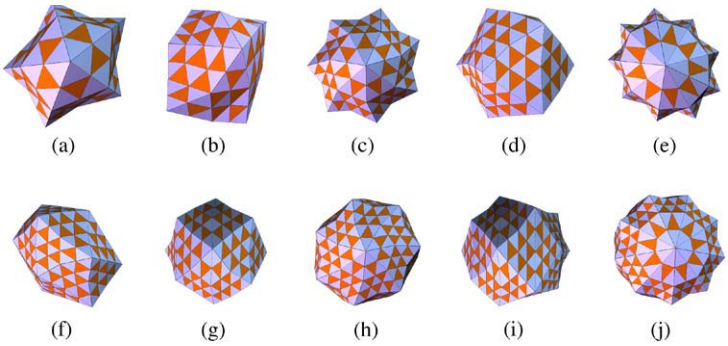

Fig. 20. Control nets for a selection of the characteristic maps of the extraordinary vertex with the valences from 7 to 22.

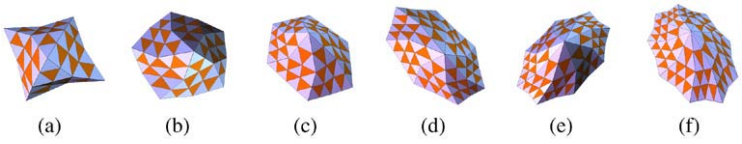

Fig. 21. Control nets for a selection of the characteristic maps of the extraordinary edges with the valences from 4 to 9.

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. [13] 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 and their subsimplices 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. Fig. 22 illustrates examples of singularities which our framework can represent. As shown in Fig. 22(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 Fig. 22(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 starting from it. The case shown in Fig. 22(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 the singular surface. Both parts have smooth surfaces as well as smooth interior, but the internal 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

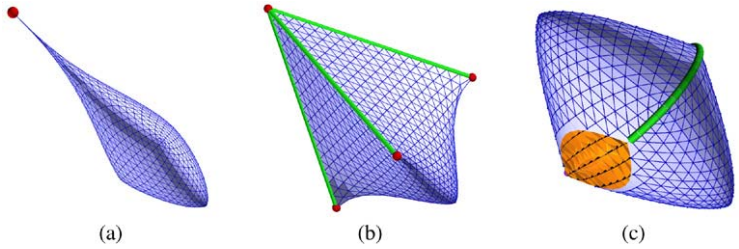

Fig. 22. Examples of singularities in manifolds. (a) A singular vertex. (b) A corner and creases. (c) A 2-manifold embedded in the 3-manifold.

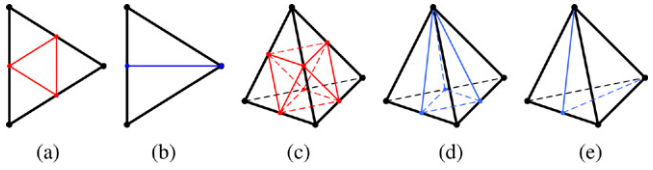

Fig. 23. Local refinement rules. (a) Red rule and (b) Green rule for local triangulation. (c) Red rule, (d) Green-III rule, and (e) Green-I rule for local tetrahedralization. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

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 Fig. 23). For the 1-simplex case, no special rule is needed. For the 2-simplex case, only the 1-ring of the adjacent simplices is affected by the Green rule (see Fig. 23(b)). For 3-simplices, the 1-ring of the adjacent simplices is split by the Green-III rule (see Fig. 23(d)), while the 2-ring of the neighboring simplices and the edge-sharing simplices are modified by the Green-I rule (see Fig. 23(e)). For an octahedral cell, we simply split it into four tetrahedra, without affecting the neighbors. 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 data in Algorithm 1 as an input. 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. To minimize the time complexity, each simplex can contain the information about its neighbors and subsimplices, which increases memory consumption exponentially during the subdivision process. We compromise both time and memory by intensive usage of hash tables and other data structure to allow fast neighbor search.

Algorithm 1 MULTI-DIMENSIONAL-SUBDIVISION.

MULTI-DIMENSIONAL-SUBDIVISION ($\mathcal{V}$, $\mathcal{C}_{\max}$, $\mathcal{S}_{\max}$)
$\{\mathcal{V} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbb{R}^3\}$
$\mathcal{C}_{\max} = \max(\mathcal{C}) = \{S \in \mathcal{C} \mid S : \mathit{maximal}\}$
$\mathcal{S}_{\max} = \max(\mathcal{S}) = \{T \in \mathcal{S} \mid T : \mathit{maximal}\}\}$
set $\mathcal{M} = \mathcal{C}_{\max} \cup \mathcal{S}_{\max}$

5.2. Complex construction

In Algorithm 2, we reconstruct the complex $\mathcal{C}$, the decomposition $\mathcal{C}_k$, and mark the type 1 non-manifold simplices according to the following process. Remember that $\rho(S)$ is 1-ring neighbors of S. After the process, newly generated subsimplices are checked to verify whether they are boundary or type 2 non-manifold simplices. The procedure is explained in Algorithm 3.

Algorithm 2 COMPLEX-CONSTRUCT.

```
COMPLEX-CONSTRUCT (V, M)
{ρ(S): 1-ring neighbor of S}
initialize each C_k as empty
for all k = 0, 1, 2, 3 do
    for all k-simplex S ∈ M do
        put S in C_k.
        for all l-subsimplex T ⊂ S with l < k do
            put T in C_k
            if T ∈ C_k', k ≠ k' then
                tag T as non-manifold type 1
            end if
            construct ρ(S) if l = 0, or 1
        end for
    end for
end for
return all C_k
```

We still need to figure out non-manifold type 3 non-manifold simplices and subsimplices of type 1 and type 2 non-manifold simplices. This has to be done at the end, because the process

Algorithm 3 FIND-BOUNDARY-AND-NON-MANIFOLD.

```
1: FIND-BOUNDARY-AND-NON-MANIFOLD (C_k)
2: for all k = 1, 2, 3 do
3:    for all new (k − 1)-subsimplex (face) T ∈ C_k do
4:       if T belongs to only one k-simplex then
5:          tag T as boundary
6:       else if T belongs to more than two k-simplex then
7:          tag T as non-manifold type 1
8:       end if
9:    end for
10: end for
```

requires type 1 and type 2 information. Here, we denote by $\mu(S)$ a number of maximal simplices that contains S. Algorithm 4 shows the steps to this process. Once the complex

Algorithm 4 FIND-TYPE-THREE-NON-MANIFOLD.

```
1: FIND-TYPE-THREE-NON-MANIFOLD (C_k)
   {μ(S): A number of maximal simplices that contains S}
2: for all k = 0, 1 do
3:    for all l-simplex T ∈ C_k with l < k − 1 do
4:       for all l'-simplex S ∈ C_k with l < l' ≤ k do
5:          if T is a subsimplex of S and dim(S) = k then
6:             increase μ(T)
7:             if μ(T) ≥ 2 then
8:                tag T as non-manifold type 3
9:             end if
10:         else if T is a subsimplex of S and dim(S) < k then
11:            if S is non-manifold then
12:               tag T as the same non-manifold type as S
13:            end if
14:         end if
15:      end for
16:   end for
17: end for
```

construction is 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, the boundary, and the manifold test. They can be safely removed from the memory once every step is done.

5.3. Subdivision process

In Algorithm 5, 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 $\mathcal{V}_{\text{new}}$ as the next level of the vertices. We follow the exactly same steps for each edge to obtain a set of new edge points, $\mathcal{E}_{\text{new}}$. Once the new vertex and edge points have been computed, we split each simplex. The process is detailed in Algorithm 6. As a result, 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 levels.

Algorithm 5 NEW-VERTEX-POINTS.

```
1: NEW-VERTEX-POINTS (V, C)
   {C = ⋃ C_k}
2: for all vertex x in V do
3:    filter ρ(x) so that it contains only the same type of
      vertices as x.
4:    choose the subdivision matrix S_x
5:    compute the vertex point v_new by S_x and the filtered ρ(x)
6:    associate v_new with x
7:    put v_new in V_new
8: end for
9: return V_new
```

Algorithm 6 SPLIT-SIMPLEX.

```
1: SPLIT-SIMPLEX (V_new, E_new, C)
2: initialize V' and C' as empty
3: for all k = 0, 1, 2, 3 do
4:    for all k-simplex S ∈ C do
5:       if k == 0 or 1 then
6:          put v_new or e_new associated with S in V'.
7:       else
8:          if S is an octahedron cell then
9:             compute the cell point c_new
10:            put c_new in V'
11:         end if
12:         split S by v_new, e_new and c_new if required
13:         put the split simplices in C'
14:      end if
15:   end for
16: end for
17: return V', C'
```

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. Fig. 24(a)–(c) show non-manifold models. In Fig. 24(a), the non-manifold region is explicitly defined by a 1-singular simplex. On the other hand, rule G-2 is used to blend the region in Fig. 24(b). A similar effect is demonstrated in Fig. 24(c). In real-world application, such as manufacturing, the 2-manifold only parts can be converted to solids by adding a certain thickness toward their normal direction. In Fig. 25(a)–(c), 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. Fig. 26(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 round holes. Finally, Fig. 27(a)–(c) show an experiment with material properties. We can apply the subdivision rules on geometric coordinates, as well as their associated material values. In this case, we assign pseudo-temperature values at the initial level, and the subdivision rules

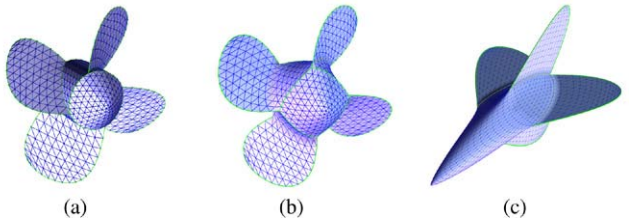

Fig. 24. Various non-manifold models by the combination of 2- and 3-manifolds.

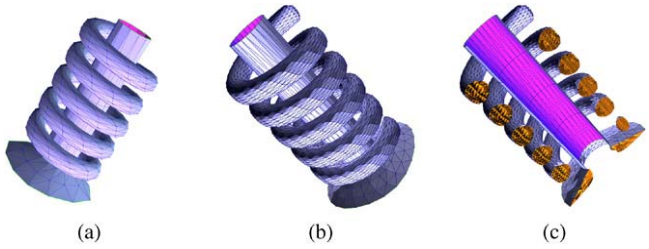

Fig. 25. A valve model with a spring.

smoothly blend them into the structure. Because of its C^1 smoothness, the result is superior to linear blending. Also, it is naturally extended to non-manifold regions.

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 on 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 subdivision scheme 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 algorithm, 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.

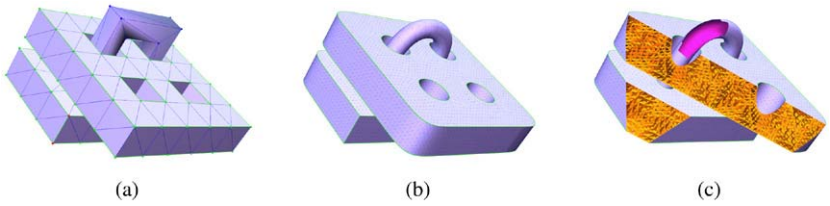

Fig. 26. A model of a mechanical part with the complex topology.

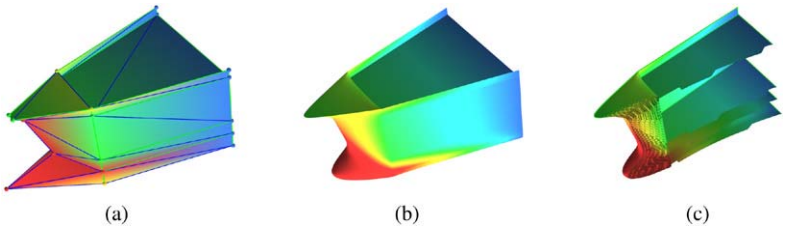

Fig. 27. A material property representation with a ship model.

Acknowledgments

The authors wish to thank Dr. 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

[1] Bajaj C, Schaefer S, Warren J, Xu G. A subdivision scheme for hexahedral meshes. The Visual Computer 2002;18:343–56.
[2] Boehm W, Farin G, Kahmann J. A survey of curve and surface methods in CAGD. Computer Aided Geometric Design 1984;1:1–60.
[3] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design 1978;10:350–5.
[4] Chang Y-S, McDonnell KT, Qin H. A new solid subdivision scheme based on box splines. In: Proceedings of solid modeling 2002. 2002. p. 226–33.
[5] Chang Y-S, McDonnell KT, Qin H. An interpolatory subdivision for volumetric models over simplicial complexes. In: Proceedings of shape modeling international 2003. May 2003. p. 143–52.
[6] Coxeter HSM. Regular Polytopes. 2nd ed. New York: The Macmillan Company; 1963.
[7] de Boor C, Höllig K, Riemenschneider S. Box splines. New York: Springer-Verlag; 1993.
[8] Doo D, Sabin M. Behaviour of recursive division surfaces near extraordinary points. Computer-Aided Design 1978;10(6):356–60.
[9] Dyn N, Micchelli CA. Using parameters to increase smoothness of curves and surfaces generated by subdivision. Computer Aided Geometric Design 1990;7:129–40.
[10] Floriani LD, Magilo P, Puppo E, Sobrero D. A multi-resolution topological representation for non-manifold meshes. In: Proceedings of solid modeling 2002. 2002. p. 159–70.
[11] Floriani LD, Morando F, Puppo E. Representation of non-manifold objects through decomposition into nearly manifold parts. In: Proceedings of solid modeling 2003. 2003. p. 304–9.
[12] Greissmair J, Purgathofer W. Deformation of solids with trivariate B-splines. In: Computer graphics forum (Proceedings of Eurographics'89). 1989. p. 137–48.
[13] 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. p. 295–302.
[14] Lasser D. Bernstein-bezier representation of volumes. Computer Aided Geometric Design 1985;2(1–3):145–50.
[15] Levin A, Levin D. Analysis of quasi uniform subdivision. Applied and Computational Harmonic Analysis 2003;15(1):18–32.
[16] Loop C. Smooth subdivision surfaces based on triangles. Master's thesis. University of Utah, Department of Mathematics; 1987.
[17] MacCracken R, Joy KI. Free-Form deformations with lattices of arbitrary topology. In: Computer graphics (SIGGRAPH'96 Conference Proceedings). 1996. p. 181–8.
[18] Micchelli C, Prautzsch H. Computing surfaces invariant under subdivision. Computer Aided Geometric Design 1987;4(4):321–8.
[19] Pascucci V. Slow growing subdivision (sgs) in any dimension: Towards removing the curse of dimensionality. Computer Graphics Forum (Proceeding of Eurographics 2002) 2002;21(3):451–60.
[20] Pasko AA, Savchenko VV. Blending operations for the functionally based constructive geometry. In: CSG 94 Set-theoretic solid modeling: Techniques and applications. 1994. p. 151–61.
[21] Popovic J, Hoppe H. Progressive simplicial complexes. In: Computer graphics (SIGGRAPH'97 Conference Proceedings). 1997. p. 217–24.
[22] Prautzsch H. Generalized subdivision and convergence. Computer Aided Geometric Design 1985;2(1–3):69–76.
[23] Prautzsch H. Smoothness of subdivision surfaces at extraordinary points. Advances in Computational Mathematics 1998;9:377–90.
[24] Reif U. Some new results on subdivision algorithms for meshes of arbitrary topology. Approximation Theory VIII 1995;2:367–74.
[25] Reif U. A unified approach to subdivision algorithms near extraordinary vertices. Computer Aided Geometric Design 1995;12:153–74.
[26] Requicha AAG, Voelcker HB. Solid modeling: A historical summary and contemporary assessment. IEEE Computer Graphics and Applications 1982;2:9–23.
[27] Schaefer S, Hakenberg J, Warren J. Smooth subdivision of tetrahedral meshes. In: Eurographics symposium on graphics processing 2004. 2004. p. 151–8.
[28] Warren J, Weimer H. Subdivision methods for geometric design: A constructive approach. Morgan Kaufmann Publisher; 2001.
[29] Wyvill B, McPheeters C, Wyvill G. Animating soft objects. The Visual Computer 1986;2(4):235–42.
[30] Ying L, Zorin D. Nonmanifold subdivision. In: Proceedings of IEEE visualization 2001. 2001. p. 325–32.
[31] Zorin D. Smoothness of stationary subdivision on irregular meshes. Constructive Approximation 2000;16:359–98.
[32] Zorin D, Schröder P. Subdivision for modeling and animation. In: SIGGRAPH 2000 course notes. 2000.
[33] Zorin D, Schröder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology. In: Computer graphics (SIGGRAPH'96 Conference Proceedings). 1996. p. 189–92.

Dr. Yu-Sung Chang is a Software Engineer and a Mathematician in Kernel Technology Department at Wolfram Research, Inc, the makers of *Mathematica*. He received his Ph.D. in Computer Science at Stony Brook University in Computer Science in 2005. He earned his B.S. degree in Mathematics from Seoul National University, Korea in 1996. He received his M.S. degree in Mathematics from Courant Institute at New York University in 2000. During 1991-97, he received an Honor Scholarship and was awarded the Honor for Excellent Graduation in 1996 from Seoul National University. In 2000, he received a Presidential Fellowship from Stony Brook University. His current projects at Wolfram Research include Scientific Visualization, Geometric Modeling, Large Dataset Processing and Multi-dimensional Analysis. He is a member of ACM and SIAM.

Dr. Hong Qin is an Associate Professor (with tenure) of Computer Science at State University of New York at Stony Brook. He received his B.S. (1986) degree and his M.S. degree (1989) in Computer Science from Peking University in Beijing, China. He received his Ph.D. (1995) degree in Computer Science from the University of Toronto. During his years at the University of Toronto (UofT), he received UofT Open Doctoral Fellowship. In 1997, Professor Qin was awarded NSF CAREER Award from the National Science Foundation (NSF). In December 2000, Professor Qin received Honda Initiation Grant Award. In February 2001, Professor Qin was selected as an Alfred P. Sloan Research Fellow by the Sloan Foundation. In June 2005, Professor Qin served as the general Co-Chair for Computer Graphics International 2005 (CGI'2005). At present, he is an associate editor for IEEE Transactions on Visualization and Computer Graphics (IEEE TVCG), and he is also on the editorial board of The Visual Computer (International Journal of Computer Graphics).