Multiresolution Heterogeneous Solid Modeling and Visualization Using Trivariate Simplex Splines

Jing Hua, Ying He, and Hong Qin

Center for Visual Computing (CVC) and Department of Computer Science
State University of New York at Stony Brook, Stony Brook, NY 11794-4400

Abstract

This paper presents a new and powerful heterogeneous solid modeling paradigm for representing, modeling, and rendering of multi-dimensional, physical attributes across any volumetric objects. The modeled solid can be of complicated geometry and arbitrary topology. It is formulated using a trivariate simplex spline defined over a tetrahedral decomposition of any 3D domain. Heterogeneous material attributes associated with solid geometry can be modeled and edited by manipulating the control vectors and/or associated knots of trivariate simplex splines easily. The multiresolution capability is achieved by interactively subdividing any regions of interest and allocating more knots and control vectors accordingly. We also develop a feature-sensitive fitting algorithm that can reconstruct a more compact, continuous trivariate simplex spline from structured or unstructured volumetric grids. This multiresolution representation results from the adaptive and progressive tetrahedralization of the 3D domain. In addition, based on the simplex spline theory, we derive several theoretical formula and propose a fast direct rendering algorithm for interactive data analysis and visualization of the simplex spline volumes. Our experiments demonstrate that the proposed paradigm augments the current modeling and visualization techniques with the new and unique advantages.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction and Motivation

New and powerful solid representations underpin the success of solid modeling and relevant applications. With the advent of ever-increasing computing power and more advanced data acquisition technologies, solid geometry has quickly gained popularity as an intuitive and natural paradigm for representing, modeling, and simulating three-dimensional real-world objects in interactive graphics, simulation, scientific visualization, and virtual environments. To date, the vast majority of popular solid modeling approaches, as well as commonly-used solid modeling systems, are built upon the following geometric foundations: constructive solid geometry (CSG), boundary representation (B-reps), and cell decomposition. When our goals are to reconstruct heterogeneous models of physical objects with continuous properties, and further model and visualize the objects, prior representations and the current state-of-the-art in solid modeling fall short in offering designers an integrated paradigm to represent complex solid geometry, arbitrary topology, and continuously-varying material attributes in a single framework simultaneously, since solid geometry and heterogeneous attributes are usually considered separately.

In sharp contrast to the previously-mentioned modeling challenges, research in visualization areas is mainly concerned with the underlying attribute fields purely for the display and visualization purpose. The representations prevalently used are oftentimes discrete in nature (e.g., voxel-based regular grids or unstructured point samples). Geometry of an underlying volume is implicitly defined in the scalar field. Historically, volumetric primitives have been based on uniform or rectilinear grid, where the data is often regularly spaced along grid lines. In the past few years, an unstructured volume representation has started to emerge as a viable modeling tool, where a tetrahedral mesh is exploited to dictate the domain of a volume [CDM*94, THJ99, CCM*00, RN00]. This type of representation is expected to be more and more popular as the irregular, adaptive 3D scanning technologies becomes commonplace. However, from a pure visualization's point of view, tetrahedra are mainly exploited as rendering primitives, i.e., they serve as a good discrete representation for visualization. This kind of tetrahedral mesh representation is only C^0. It is less suitable for modeling continuously-varying material attributes without resorting to approximation.

To satisfy the modeling requirement of high-order continuity, volumetric modeling based on splines such as B-splines or NURBS [RE99, HQ01, SPS01, MC01] appears to be more appropriate. Nonetheless, modeling with B-splines or NURBS has severe shortcomings. Its modeling scope is extremely constrained in term of geometric, topological, and attribute aspects. First, B-spline and NURBS are defined over a regular, tensor-product domain. A single

B-spline or NURBS can not represent volumes of arbitrary topology without patching or trimming operations. Furthermore, patching multiple B-splines or NURBS to form arbitrary topology is not easy to control at all. Second, tensor-product splines are essentially smooth everywhere. It is difficult to model high-frequency features. Third, when refining a region of interest in a tensor-product spline patch, it will introduce too many extra degrees of freedom in other less-interesting regions nearby in order to retain its regular structure. Attractive properties such as local adaptivity and multiresolution are rather difficult to achieve.

To overcome the above difficulties in both modeling and visualization, in this paper we propose an integrated approach for representing, modeling, and rendering of multi-dimensional, physical attributes across any volumetric objects. We consider modeling both the geometry and the attributes carried by the volume. In particular, we develop a trivariate simplex spline model that is defined over a tetrahedral decomposition of any 3D domain. The modeled volume can be of complicated geometry and arbitrary topology. Our goal is to demonstrate that the trivariate simplex spline is a promising primitive for both visualization and modeling tasks, especially for representing and visualizing heterogeneous models of physical objects and their material properties. Our model makes use of a more general and flexible tetrahedral domain and offers a compact continuous representation at the same time. It unifies geometry and attribute properties over domains of arbitrary topology. It is possible to represent a complicated heterogeneous object with a single trivariate simplex spline without any additional operations of trimming or patching, while the geometry of the object is explicitly represented by the spline. The trivariate simplex spline based representation offers a single, compact analytic representation, because it is a piecewise polynomial of the lowest possible degree and the highest possible continuity everywhere across the entire tetrahedral domain.

For example, given a degree n for the trivarite DMS spline which use simplex spline as basis functions, the representation scheme can achieve C^{n-1} continuity over smooth regions. Meanwhile, by placing control points and their associated knots in certain locations, variable continuity is readily accomplished including a C^0 continuity that defines sharp features. This property is ideal for data reduction when reverse-engineering a continuous spline model from the discretized data inputs.

Since we develop the trivariate simplex spline for both solid geometry and the material attributes associated with the solid geometry simultaneously, it facilitates the modeling of heterogeneous objects. The feature-sensitive fitting algorithm that we develop can reconstruct a more compact trivariate simplex spline from a structured or arbitrary unstructured volume. It reconstructs the geometry and the associated material attributes simultaneously. The C^{n-1} continuity and C^0 continuity can both be modeled with ease. Such flexibility allows us to model continuously-varying material distribution. It may be noted that, in our framework we use time-varying knots instead of fixed knots, which offers more freedom and improves accuracy for approximation. The knots are explicitly and automatically determined by optimizing a specific objective function. Unlike frequently used B-spline, our new algorithm can avoid the complicated procedures of trimming and patching when dealing with these cases.

This representation can also enable the strong multiresolution modeling capability through interactively subdividing any region of interest, allocating more knots and control points accordingly. Other modeling advantages such as local adaptivity and free-form deformation are also apparent by means of displacing control points, direct manipulation of control coefficients, and adjustment of free knots. In addition, based on the simplex spline theory, we derive several theoretical formula and propose a fast direct rendering algorithm for interactive data analysis and visualization of the simplex spline volumes. When visualizing this type of solids, resampling or interpolation process is no longer necessary at all. It (including position, derivative, etc.) can be evaluated anywhere analytically and computed efficiently for volume rendering.

We conduct extensive experiments and our empirical results demonstrate that the proposed paradigm significantly augments the currently existing techniques in modeling and visualization research areas. The applications of our techniques are diverse, including solid modeling of arbitrary topological types, material editing and reconstruction, volume simplification, and data exploration and visualization in bio-medical and geological fields.

2. Related Work

Research on volume modeling using B-splines or NURBS has received much attention in the modeling community in recent years. Raviv and Elber [RE99] presented a 3D interactive sculpting paradigm that employed a set of scalar uniform trivariate B-spline functions as object representations. Schmitt and Pasko [SPS01] presented an approach for constructive modeling of FRep solids defined by real-valued functions using 4D uniform rational cubic B-spline volumes as primitives. Hua and Qin [HQ01, HQ02] presented a haptics-based direct manipulation and exploration of scalar B-spline volumes. Martin and Cohen [MC01] presented a completed mathematical framework for representing and extracting volumetric attributes using trivariate NURBS.

In the visualization community, volume modeling and rendering via tetrahedral mesh has recently gained more popularity as well. Primarily, researchers are interested in constructing or using tetrahedral mesh of volume dataset to achieve better rendering effects. Cignoni *et. al* [CDM*94] proposed a multiresolution model for the representation and visualization of unstructured volumetric datasets based on a decomposition of the three-dimensional domain into tetrahedra. Later, they presented a tetrahedral mesh simplification approach based on accurate error evaluation [CCM*00]. Roxborough and Nielson [RN00] presented a method for visualization of freehand collected three-dimensional ultrasound data based on adaptive, progressive construction of the tetrahedral mesh. Yao and Taylor [YT00] adopted a tetrahedral mesh structure to represent anatomical structures. They proposed an efficient and automatic algorithm to construct a tetrahedral mesh from contours in CT images. A rich body of previous work on tetrahedral meshes suggest that a simplicial complex is potentially promising to serve for both visualization and modeling.

Multivariate simplex splines for approximation theory have been extensively investigated in mathematical science for many years. Motivated by an idea of Curry and Schoenberg for a geometric interpretation of univariate B-splines, de Boor [dB76] first presented a brief description of multivariate simplex splines. Since then, their theoretical aspects have been explored extensively. From the point of view of blossoming, Dahmen *et. al* [DMS92] proposed triangular B-splines. In sharp contrast to the theoretical advances, the

application of multivariate simplex splines has been severely under-explored. Greiner and Seidel [GS94] demonstrated the practical feasibility of multivariate B-spline algorithms in graphics and shape design. Pfeifle and Seidel proposed a faster evaluation technique for quadratic bivariate DMS-spline surfaces [PS94] and demonstrated the fitting of triangular B-spline surfaces to scattered data through the use of least squares and optimization techniques [PS95]. Qin and Terzopoulos [QT97] presented dynamic triangular NURBS, a freeform shape model. Most recently, He and Qin [HQ04] presented a new approach for reconstructing a triangular B-spline surface from a set of scanned 3D points. To the authors' best knowledge, there are no existing research work, which applies multivariate simplex splines to represent solid geometry, model its heterogeneous material attributes, and reconstruct continuous volumetric splines from discretized volumetric inputs via data fitting.

3. Trivariate DMS-spline Volumes

Analogous to a tensor-product, trivariate B-spline volume [RE99, MC01], we can instead use simplex splines to model volumetric objects. To motivate our rationales, we have detailed some major advantages of multivariate simplex spline volumes over conventional tensor-product B-spline volumes in Section 1.

3.1. Trivariate Simplex Splines

Throughout this paper, we employ a trivariate simplex spline to represent and extract both solid geometry and its volumetric attributes. Now we shall review the formulation of the trivariate simplex splines and summarize their analytic and geometric properties.

A degree n trivariate simplex spline, $M(\mathbf{x}|\mathbf{x}_0, \cdots, \mathbf{x}_{n+3})$, can be defined as a function of $\mathbf{x} \in \mathbb{R}^3$ over the half open convex hull of a point set $\mathbf{V} = [\mathbf{x}_0, \cdots, \mathbf{x}_{n+3})$, depending on the $n+4$ knots $\mathbf{x}_i \in \mathbb{R}^3, i = 0, \cdots, n+3$. The basis function of trivariate simplex splines may be formulated recursively, which facilitates point evaluation and its derivative and gradient computation. When $n = 0$,

$$M(\mathbf{x}|\mathbf{x}_0, \cdots, \mathbf{x}_3) = \begin{cases} \frac{1}{|\mathrm{Vol}_{\mathbb{R}^3}(\mathbf{x}_0, \cdots, \mathbf{x}_3)|}, & \mathbf{x} \in [\mathbf{x}_0, \cdots, \mathbf{x}_3), \\ 0, & \text{otherwise,} \end{cases}$$

and when $n > 0$, select four points $\mathbf{W} = \{\mathbf{x}_{k_0}, \mathbf{x}_{k_1}, \mathbf{x}_{k_2}, \mathbf{x}_{k_3}\}$ from $\mathbf{V}$, such that $\mathbf{W}$ is affinely independent, then

$$M(\mathbf{x}|\mathbf{x}_0, \cdots, \mathbf{x}_{n+3}) = \sum_{j=0}^{3} \lambda_j(\mathbf{x}|\mathbf{W}) M(\mathbf{x}|\mathbf{V} \setminus \{\mathbf{x}_{k_j}\}), \quad (1)$$

where $\sum_{j=0}^{3} \lambda_j(\mathbf{x}|\mathbf{W}) = 1$ and $\sum_{j=0}^{3} \lambda_j(\mathbf{x}|\mathbf{W})\mathbf{x}_{k_j} = \mathbf{x}$.

The directional derivative of $M(\mathbf{x}|\mathbf{V})$ with respect to a vector $\mathbf{v}$ is defined as follows:

$$D_{\mathbf{v}}M(\mathbf{x}|\mathbf{V}) = n \sum_{j=0}^{3} \mu_j(\mathbf{v}|\mathbf{W}) M(\mathbf{x}|\mathbf{V} \setminus \{\mathbf{x}_{k_j}\}), \quad (2)$$

where $\mathbf{v} = \sum_{j=0}^{3} \mu_j(\mathbf{v}|\mathbf{W})\mathbf{x}_{k_j}$ and $\sum_{j=0}^{3} \mu_j(\mathbf{v}|\mathbf{W}) = 0$. In the interest of space, more theoretical discussions can be found elsewhere.

3.2. Trivariate DMS-spline Volumes

In [DMS92], Dahmen, Micchelli and Seidel presented a multivariate B-spline scheme, called DMS-spline, based on blending functions and control vectors. The surface scheme is also called triangular B-spline, which has been studied in [GS94, PS94, PS95, QT96, SV94].

We apply the trivariate DMS-spline to represent both solid geometry and its associated physical attributes in this paper. Let Ω be an arbitrary proper tetrahedralization of $\mathbb{R}^3$ or some bounded domain $D \subset \mathbb{R}^3$. Here, "Proper" means that every pair of domain tetrahedra are disjoint, or share exactly one vertex, one edge, or one face.

To each vertex $\mathbf{t}$ of the tetrahedralization, we assign a knot cloud, which is a sequence of points $[\mathbf{t}_0, \mathbf{t}_1, \cdots, \mathbf{t}_n]$, where $\mathbf{t}_0 \equiv \mathbf{t}$. For every tetrahedron $I = (\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s})$, we require

- all the tetrahedra $[\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l]$ with $i + j + k + l \leq n$ are non-degenerate.
- the set

$$interior(\cap_{i+j+k+l \leq n}[\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l]) \neq \varnothing \quad (3)$$

is not empty.

- if I has a boundary triangle, the knots associated to the boundary triangle must lie outside of Ω.

We then define, for each tetrahedron I and $i + j + k + l = n$ (in the following, we use β to denote 4-tuple (i, j, k, l), the knot sets

$$V_{\beta}^{I} = [\mathbf{p}_0, \cdots, \mathbf{p}_i, \mathbf{q}_0, \cdots, \mathbf{q}_j, \mathbf{r}_0, \cdots, \mathbf{r}_k, \mathbf{s}_0, \cdots, \mathbf{s}_l]. \quad (4)$$

The basis functions of normalized simplex splines are then defined as

$$N_{\beta}^{I}(\mathbf{u}) = |d(\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l)| M(\mathbf{u}|V_{\beta}^{I}), \quad (5)$$

where $|d(\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l)|$ is six times of the volume of $(\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l)$. Like the ordinary tensor-product B-spline, a trivariate simplex spline volume of degree n over arbitrary tetrahedral domain is the combination of a set of basis functions with control vectors $\mathbf{c}_{\beta}^{I}$:

$$\mathbf{s}(\mathbf{u}) = \sum_{I \in \Omega} \sum_{|\beta|=n} \mathbf{c}_{\beta}^{I} N_{\beta}^{I}(\mathbf{u}). \quad (6)$$

Like B-splines, the nonnegative basis functions of simplex spline volume sum to unity. They have a number of nice properties, such as the convex hull property, local support, affine invariance. Shape design based on trivariate DMS-spline volume includes the specification of a domain tetrahedralization, knot sequences, and control points to generate an initial shape. The initial shape is then refined into the final desired shape through interactive adjustment of domain tetrahedralization, control points, and knots. The geometric flexibility of simplex spline volumes provides great power on its shape editing. Figure 1 show a cubic trivariate DMS-spline solid corresponding to a domain with a single tetrahedron. Note that our trivariate DMS volumes represent not only boundary geometry, but also interior solid geometry. They can represent physical or material attributes over the entire solid as well. Figure 1 shows scaled tetrahedra of the solid in order to emphasize its non-empty solid interior geometry.

For a general trivariate DMS-spline volume, each domain tetrahedron I has its own set of control points $\mathbf{c}_{\beta}^{I}$. However, in this paper, we consider a more restricted class of volumes by sharing respective control points along common triangles of two adjacent tetrahedra in parametric tetrahedralization. Splines with shared control points have a direct visual effect in geometric and solid modeling. More importantly, as proved by Dahmen et. al [DMS92], a degree
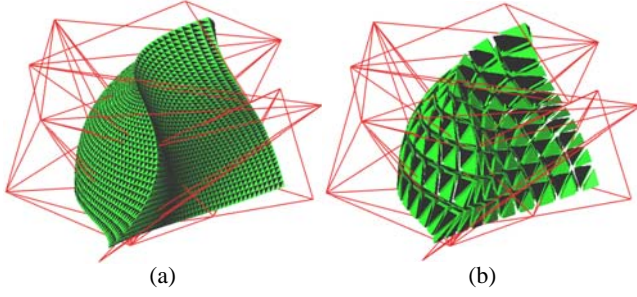
Figure 1: *(a) A cubic trivariate DMS-spline solid corresponding to a single tetrahedral domain with 20 control points. (b) The tetrahedra of the designed solid object are scaled to show the interior of the solid.*

n trivariate DMS-spline with shared control points can be evaluated with the efficiency of a degree $n-1$ spline.

$$\mathbf{s}(\mathbf{u}) = \sum_{I \in \Omega} \sum_{|\beta|=n-1} \hat{\mathbf{c}}_\beta^I(\mathbf{u}) N_\beta^I(\mathbf{u})$$

where

$$\hat{\mathbf{c}}_\beta^I(\mathbf{u}) = \sum_{m=0}^{3} \mathbf{c}_{\beta+e^m}^I \lambda_m(\mathbf{u}|\mathbf{p}_i, \mathbf{q}_j, \mathbf{r}_k, \mathbf{s}_l)$$

and $e^m = (\delta_{i,m})_{i=0}^3$, $m = 0, 1, 2$ as the coordinate vectors.

This property can significantly improve the software system for rendering a DMS-spline volume. It also implies that the knots $\{\mathbf{t}_{i,n} | i \in \mathbf{N}\}$ has no effect on the value of $\mathbf{s}(\mathbf{u})$.

Note that for our trivariate DMS spline volumes, the used tetrahedral domain is not necessary to undergo a separate parameterization process, because it is already conforming to the three-dimensional physical domain for material attributes. The *generalized* control vectors are now $(n+3)$ vectors, including control points (p^x, p^y, p^z) for the solid geometry, and control coefficients $(g^1, \cdots, g^n)$ for the attributes, where n denotes the number of attributes associated with the geometry.

3.3. Coupling Solid Geometry and Physical Attributes

In our framework, we consider a control coefficient (and possibly other vector-based quantities with n components) is associated with a corresponding control point and evaluated with the geometry simultaneously based on the same tetrahedral domain. In this way, we shall generalize DMS spline technique from geometric domains to visual or material domains. Typical material-oriented examples include: mass, damping, stiffness (such as strain and stress), and displacement for solid physics; or density, velocity, and pressure for fluid mechanics, etc. These attributes may be assigned at the control points or fitted by using both control vectors and knots. Other commonly-used visual information include: color, texture, intensity, opacity, transparency, etc. This flexibility permits our paradigm to be employed in a wide variety of applications involving continuous domains, such as finite element analysis, virtual sculpting, and computational fluid simulation.

Consider that control coefficients $\mathbf{g}_\beta^I$ are associated with control points $\mathbf{p}_\beta^I$ ($\mathbf{g}_\beta^I$ may be a multi-dimensional vector). Besides solid geometry, a material (e.g., density) function over $\mathbf{s}(\mathbf{x})$ can be simultaneously defined as

$$\begin{bmatrix} \mathbf{g} \\ \mathbf{s} \end{bmatrix}(\mathbf{x}) = \sum_i \begin{bmatrix} \mathbf{g}_\beta^I \\ \mathbf{p}_\beta^I \end{bmatrix} N_\beta^I(\mathbf{x}), \tag{7}$$

where $\mathbf{g}$ can be color, texture, mass, temperature, or other visual or material functions as stated above. Visual and material modeling is indispensable for computerized virtual environments. Besides commonly-used material distributions in classical mechanics, it can be generalized to heat transfer, electricity, and beyond. The diverse set of these novel solid models are unique and invaluable because they can potentially unify geometric, topological, kinematic, material, and dynamic properties.

The field attributes can be modified by directly changing the control coefficients stored at the control points. In addition, by moving the control points, we can perform FFD of the underlying geometric space. As a result, this procedure deforms the field properties and provides an alternate interaction mechanism. The geometric space over which the field is defined can be of arbitrary topology, which adds further shape editing flexibility.

4. Multiresolution Modeling

Based on the flexible simplex splines, we can easily achieve multiresolution modeling of heterogeneous solid objects. The multiresolution capability is achieved by interactively subdividing any region of interest and allocating more knots and control vectors accordingly.

4.1. Interactive Tetrahedralization

We use Delaunay tetrahedralization to construct a tetrahedral domain, which serves as the tetrahedral domain of the trivariate simplex spline. In fact, the scalar simplex splines lend itself well appropriate for an unstructured grid volume representation.

For constructing Delaunay tetrahedralization for a 3D point set, we make use of the incremental flip algorithm proposed in [Joe91, ES92]. The procedure to update a tetrahedralization is to insert one new vertex at each step. Then, a sequence of actions are applied to modify the tetrahedralization locally. Each flip action consists of replacing one, two, or three adjacent tetrahedra with four, three, or two new tetrahedra, respectively. When the new vertex lies either on a triangular face or on an edge of the current tetrahedralization, other face flips may be needed. When performing Delaunay tetrahedralization, the user have an option to enforce the quality constraints (including volume of tetrahedron, etc.).

The user can also interactively specify boundary constraints (such as points, edges, polygons, or cells). We implement the boundary conforming Delaunay tetrahedralization [She97], which tries to recover the missing boundary edges of the piecewise linear complexes from its current Delaunay tetrahedralization by inserting new points.

4.2. Geometric Editing Using Control Points

With this flexible modeling technique, we can easily design solid geometric shapes with sharp features and associated high frequency material properties. Since the control vectors include the position

of control points and associated control coefficients, editing of control points and/or control coefficients offer more powerful modeling capabilities.

Based on our experiments, we observe that associating control points to a domain in such a way: locating control points at the edges of tetrahedra of the domain, can well preserve the shape and features represented by the original domain. The difference between the geometry of the domain and the resulting DMS solid object is very small.

Moving control points around can lead to a desired deformation of the underlying model easily. Figure 2 shows an example. Figure 2(a) shows the domain of the cubic DMS solid, which is a torus. The user moves the control points to the new locations as shown in the solid mesh in Figure 2(b). Six very sharp corners are formed on the original solid object.
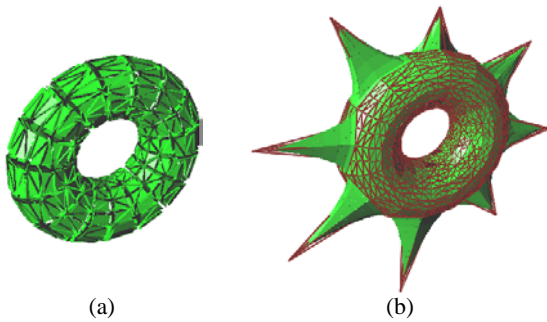

(a) (b)

Figure 2: *(a) The domain is a torus. (b) The deformation of the object is achieved by moving the control points to the new locations. The mesh shows a part of control nets, while the flat shaded surface indicates the resulting solid objects.*

Usually when the number of control points is very large, it is tedious and relatively difficult to manipulate the control points individually. In our system we provide Free-Form Deformation (FFD) tools to allow users to move control points. Figure 3 shows another example, where the domain is 5 separated tori as shown in Figure 3(a), and the DMS solid is quadratic. After the initial control points are generated, the user employ the bending FFD tool to relocate the control points to achieve the design shown in Figure 3(b). Note that, it is almost impossible for B-splines to model such objects whose domain is of non-trivial topology.

(a) (b)

Figure 3: *(a) The domain consists of 5 separated tori, which is of non-trivial topology. (b) The deformation of the object is achieved by bending the control net.*

Figure 4 shows another example. The user first builds a 3D piecewise linear complex as shown in Figure 4, then our system generates

its quality conforming Delaunay mesh. Badly shaped elements from the mesh are eliminated and replaced with well shaped ones. This is done by automatically inserting additional points into current mesh. From Figure 4(b), we can see that more knots are generated around feature edges. By associating control points with the defined domain, all the features are preserved in the resulting DMS solid of degree 4. Please refer to Figure 4(d).

(a) (b)

(c) (d)
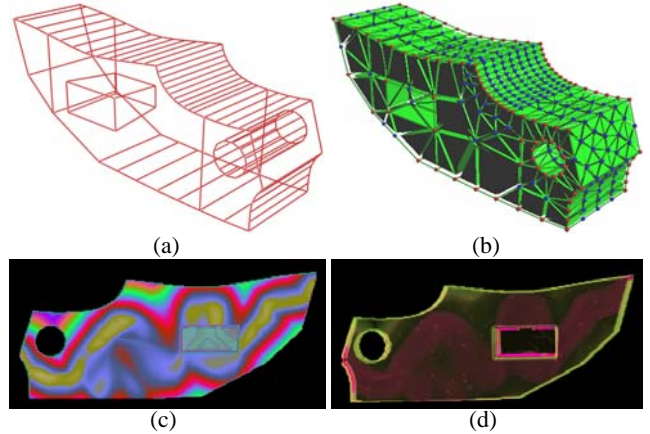
Figure 4: *(a) The piecewise linear boundary constraints that the user specifies. (b) The multiresolution tetrahedralization conforming to the piecewise linear boundary constraints. (c) The color map of material distribution of the designed object. (d) Volume rendering of the designed object, where we can see that all the geometric shape features are preserved.*

4.3. Attribute Editing Using Control Coefficients or Control Points

Besides modeling solid geometry, the user can easily edit associated physical or material attributes as well. This is achieved by editing control coefficients at the corresponding control points, because the underlying material distribution is in fact a trivariate simplex spline defined over the same tetrahedralization. In the most simple form, the (multi-dimensional) control coefficients can be in one-to-one correspondence to their control-point counterparts which are used to define arbitrarily curved solid geometry over the same parametric domain (which is the initial tetrahedralization). However, local refinements for material modeling are frequently desirable in order to represent high-frequency features in the material domain.

Our system allows users to interactively sketch skeletons. Each skeletal element is then associated with a locally defined implicit function; individual functions are blended using a polynomial weighting function that can be controlled by the user. After specifying these, the scalar control coefficients of the DMS-spline solid are assigned as the evaluations of the blending of field functions g_i of a set of skeletons $s_i(i = 1, \cdots, N)$ at the positions of corresponding control points,

$$f(x,y,z) = \sum_{i=1}^{N} g_i(x,y,z), \qquad (8)$$

where the skeletons s_i can be any geometric primitive admitting a well defined distance function: points, curves, parametric surfaces,

simple volumes, etc. The field functions g_i are decreasing functions of the distance to the associated skeleton,

$$g_i(x,y,z) = G_i(d(x,y,z,s_i)), \qquad (9)$$

where $d(x,y,z,s_i)$ is the distance between (x,y,z) and s_i, and G_i can be defined by pieces of polynomials or by more sophisticated anisotropic functions. Therefore, the user may enforce global and local control of an underlying scalar field in three separate ways: (1) defining or manipulating of the skeleton, (2) defining or adjusting those implicit functions defined for each skeletal element, and (3) defining a blending function to weight the individual implicit functions.

Figure 4 shows an example, where the user uses a mouse to freely sketch a curve inside the engineering part and defines an implicit function, $f(d) = \frac{1}{1+d^2}$, to associate with the skeleton. Then the scalar control coefficients at the control points are assigned according to the skeleton function. The resulting material distribution is shown in Figure 4(c)(d).

As shown in Figure 5, the user sketches a solid modeling conference logo, "SM", with free hand inside a domain, which is a torus. After associating control points with the torus domain and assigning control coefficients according to the distance function to the sketched curve logo, the resulting geometry and its associated scalar field can be easily generated. In this example, we use a trivariate DMS-spline of degree 4. Figure 5(a) shows the volume rendering of the entire object. We can clearly see the resulting geometry and its associated attribute field. In Figure 5(b), the shaded surface shows the isosurface of the attribute field of the solid torus, which is generated using Marching Tetrahedra Method that we developed (We will discuss it in Section 6.2).
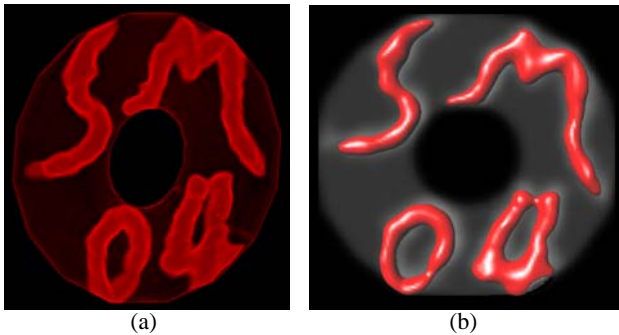

(a)   (b)

Figure 5: *(a) Editing attributes inside the tetrahedral domain of the torus model. From the direct volume rendering, we can clearly see the geometric boundary of the torus and the highlighted attributes. (b) Marching tetrahedra rendering of an isosurface of the resulting attribute field over the torus model. The greyscale background shows the torus geometry.*

In Figure 6, the user is trying to design a digest system of the bunny model. The user draws a ellipsoid like shape to represent its stomach and two pipes adjacent to it (Illustrated in Figure 6(a)). Figure 6(b) shows the tetrahedralization of the bunny, which serves as the domain. In this example, we use a quadratic trivariate DMS-spline. After associating control points with the domain, the resulting geometry and its associated attribute field can be easily generated. Please see Figure 6(c)(d).

(a)   (b)

(c)   (d)
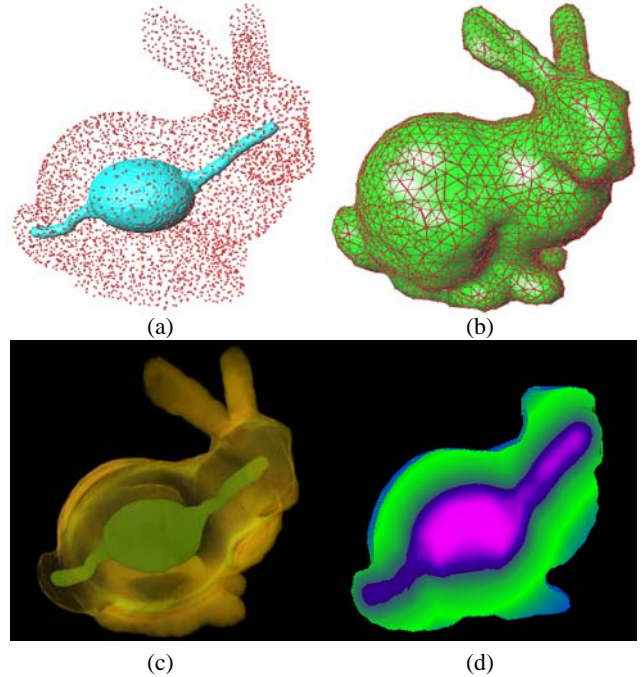
Figure 6: *(a) Define a skeleton inside the Stanford bunny model for material editing. (b) The generated tetrahedral mesh of the Stanford bunny model. (c) Direct volume rendering of the resulting attribute field over the bunny model. (d) The color map of one slice of the bunny model shows the resulting attribute distribution.*

By aligning free knots along the feature lines/faces in the domain, sharp features in both geometric and attribute fields can be modeled. As we state in Section 4.2, associating control points to the edges of tetrahedra of a domain can also well preserve the shape and features represented by the original domain. Let us see Figure 4 as an example. All the geometric features shown in the original domain are well preserved in the final resulting geometric object as shown in Figure 4(d).

In addition, by moving the control points, we can perform FFD of the underlying geometric object. As a result, this procedure deforms the associated attribute field. This is an alternate interaction mechanism for editing attributes.

5. Feature Sensitive Data Fitting

To model data or attribute over the simplex spline based volume, it is much more desirable to have a fitting tool in addition to modeling tools as presented above. In this section, we propose a feature sensitive data fitting algorithm. That means the fitting algorithm can represent the data over the volume accurately and recover all the features with as few control vectors and bases as possible. Also the geometry of the volume is recovered simultaneously. Note that in the following section, we mainly discuss how to fit an unstructured volume. In fact, it can be any format, consisting of a set of points with associated attributes (i.e., $(x,y,z,d_1,\cdots,d_n)$). Figure 7 shows the spx model from the fluid dynamics research community. In order better to show the advantages of our fitting algorithm. We upsample the spx dataset from 2896 sampling points to 15832 points.
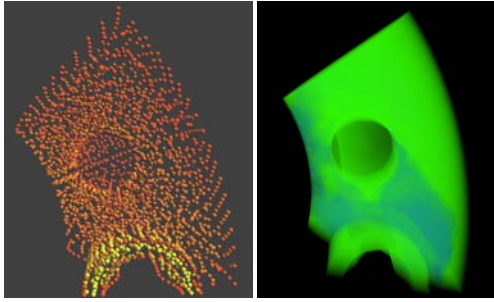
Figure 7: *(a) The point view of the spx dataset, where the color indicates the density difference. (b) Volume rendering of the spx dataset.*

5.1. The Fitting Algorithm

The problem of fitting volume data can be stated as follows: given a set $P = \{\mathbf{p}_i\}_{i=1}^{m}$ of points $\mathbf{p}_i = (x_i, y_i, z_i, d_i) \in \mathbb{R}^4$, find a trivariate DMS volume $s : \mathbb{R}^3 \rightarrow \mathbb{R}^4$ that approximates P. Unlike the existing fitting algorithms with parametric representations which usually find a one-to-one mapping between the data points and the points in the parametric space, our method skips this parameterization procedure. As stated before, we first build up a tetrahedralization parametric domain which is close to the original geometry of the to-be-fitted dataset. This domain serves for fitting both geometry and attributes. We use the position $(x_i, y_i, z_i)$ of the data point $\mathbf{p}_i$ as its parametric value. Therefore, we need to minimize the following objective function:

$$\min E = \sum_{i=1}^{m} (\mathbf{p}_i - \mathbf{s}(x_i, y_i, z_i))^2. \tag{10}$$

Our fitting algorithm treats both control vectors and knots as free variables. In this way, it can greatly reduce the approximation error given the same number of control vectors and knots. The tight coupling of both geometry and attributes during data fitting enables further editing on the continuous representation by means of moving control points, FFD, or changing control coefficients as we discussed in the Section 4, even after the fitting process is done.

We first give an overview of our fitting algorithm. Then we will discuss some related issues in detail.

1. Create a tetrahedral domain for the entire volume domain, which fully contains the fitted volumetric object;
2. Solve Equation 10 by treating control vectors as free variables.
3. for each node $\mathbf{t}_i$ of the tetrahedralization, if the fitting error in its 1-ring neighboring tetrahedra is larger than a user-specified value, solve Equation 10 by treating the knots associated to $\mathbf{t}_i$ as free variables.
4. For each tetrahedron, if its fitting error is larger than a threshold ε, then subdivide it into four tetrahedra and repeat (2-4) until the fitting error of every tetrahedron is less than ε

This algorithm will not stop until the fitting error in each tetrahedron is less than the user-specified error bound. Then the discrete point set with the associated attribute field is converted to a continuous spline-based volumetric implicit function, which can be evaluated at arbitrary sampling resolution and rendered with the direct volume rendering or the Marching Tetrahedra algorithm. The fitting algorithm is guaranteed to converge since the number of sampling points is finite. The number of tetrahedra needed for fitting at the desired accuracy depends on the user-specified error bound and the datasets.

5.2. Finding a Good Initial Tetrahedralization

As we can see from the fitting algorithm, a good initial basis will save a lot of time of performing recursive refinement and local fitting. More important, it can help to preserve geometric features of the original volume datasets. Essentially, there should be more primary knots distributed in the region containing features. Therefore, the dataset will undergo a preprocessing stage before fitting.

First we have to find the proper tetrahedralization of the point sets. We perform the Delaunay tetrahedralization of the point set. Now we have to consider how to remove those tetrahedra which are outside the actual object. We place a ball at every point, whose radius is equal to the shortest distance from this point to its one-ring neighbors. Then, we perform the union of balls to obtain an occupancy map, which can roughly indicates the boundary of the actual object. Figure 8(a) illustrates the occupancy map of the point set of the spx dataset. Third, we check each tetrahedron to see if all the center points of its six edges are inside this occupancy map. If not, this tetrahedron is clipped away. Figure 8(b) shows the occupancy map and the Delaunay tetrahedralization together, where we can see some of tetrahedra are outside the occupancy map. From Figure 8(c), we can see that all the outside tetrahedra are removed and the final tetrahedralization of the point set is obtained. In this paper, we consider the fact that the real datasets to be fitted are usually densely sampled. This algorithm does not work well for very scattered datasets. Note that, this preprocessing is to produce a tetrahedral domain, not to generate the tetrahedralization of the object. The domain tetrahedralization should be much coarser than the tetrahedralization of the object as shown in Figure 8(c).

(a)	(b)	(c)
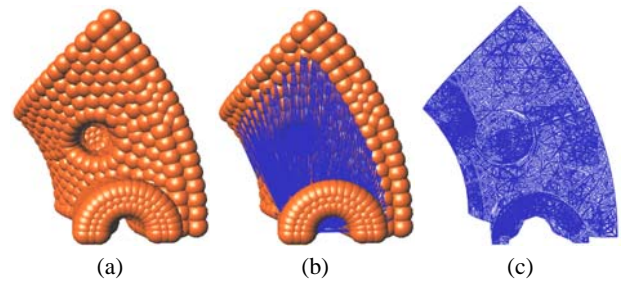
Figure 8: *(a) Occupance map of the point set. (b) The Delaunay tetrahedralization of the point set. (c) The final tetrahedralization after removing the outside tetrahedra.*

In order to let the generated tetrahedral domain faithfully reflect the nature of the object, the features should be considered in tetrahedralization. Essentially, we have two types of features since we consider both the geometry and physical attributes. Geometric features are one category and field features belongs to another category. Geometric features mean those regions where C^0 continuity occurs.

We use an efficient algorithm to classify the boundary vertices. The boundary vertices are identified as corner vertex, curve vertex, and general boundary vertex. The classification algorithm is based on the solid angle at each vertex. The solid angle α_i of

tetrahedron $I(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ at vertex $\mathbf{p}_0$ is defined to be the surface area formed by projecting each point on the face not containing $\mathbf{p}_0$ to a unit sphere centered at $\mathbf{p}_0$. An equation for the computation of solid angle α_i is given by Liu et al.[LJ94]: $\sin \frac{\alpha_i}{2} = 12v / \sqrt{\prod_{1 \leq i < j < \leq 3}[(l_{i0} + l_{j0})^2 - l_{ij}^2]}$, where v is the volume of T, and l_{ij} is the length of the edge connecting vertices $\mathbf{p}_i$ and $\mathbf{p}_j$. The solid angle α at the vertex $\mathbf{p}_0$ is the sum of the solid angles in all tetrahedra incident at $\mathbf{p}_0$. For an interior vertex, the solid angle is 4π, while the boundary vertex is less than 4π. We identify the type of $\mathbf{p}_0$ as follows [HK03]:

$$
\left\{
\begin{array}{l}
\mathbf{p}_0 \text{ is a corner boundary vertex if } \alpha \leq \frac{\pi}{2} \text{ or } 4\pi - \alpha \leq \frac{\pi}{2} \\
\mathbf{p}_0 \text{ is a curve boundary vertex if } \frac{\pi}{2} < \alpha \leq \frac{3\pi}{2} \text{ or } \frac{\pi}{2} < 4\pi - \alpha \leq \frac{3\pi}{2} \\
\text{otherwise, } \mathbf{p}_0 \text{ is a general boundary vertex}
\end{array}
\right.
$$

Once the vertices are classified, we can extract feature lines from those corner boundary vertices and curve boundary vertices. Starting from a corner vertex, $\mathbf{p}_i$, we can link adjacent edge vertices by examining each of its neighbor vertices, which are either corner boundary vertices or curve boundary vertices, and see if they have similar normal orientation

$$\|\mathbf{N}(\mathbf{p}_i) - \mathbf{N}(\mathbf{p}_j)\| \leq A$$

for some angular threshold A. Once the link is established, we start to traverse the neighbor one, $\mathbf{p}_j$, until it reaches another corner boundary points or it cannot proceed. The sharp feature lines of Spx dataset are shown in Figure 9(a) in red.

We sample the feature lines based on their curvature and a user-specified sampling rate. As a result, all these feature lines are well preserved as the piecewise linear boundary of the initial tetrahedralization to represent the shape of the to-be-fitted object. We also check the curvature of boundary surfaces according to the classified general boundary vertices to determine the placement of primary knots on those boundary surfaces for the initial tetrahedralization. For the interior vertices, we can distribute the initial primary knots according to the density of the original sampling points of the dataset. After all the primary knots have been determined, we can perform constrained tetrahedralization to obtain the initial tetrahedralization. Figure 9(b) shows the constructed initial tetrahedralization.

Figure 9: *(a) Geometric features of the spx dataset. (b) The finally constructed initial tetrahedralization.*

So far the initial tetrahedralization only considers the explicit geometric features of the object. Features of its attribute field have not been taken into account. In a nutshell, field features include

critical points, field discontinuity, high frequency region, etc. Ideally, these features should be detected and explicitly represented for constructing a feature-based tetrahedralization. However, it is still a challenging research problem in computer vision to find all the high level features such as surfaces, field discontinuity, and so on. That research topic is beyond the scope of this paper. In practice we simply employ a very straightforward method. We just allocate more primary knots in the region where the deviation of gradients is high. Based on the tetrahedralization constructed purely according to the geometric information, we calculate the deviation of gradients for each tetrahedron. If the deviation is greater than a threshold, we insert a new vertex into the tetrahedron, where the gradient magnitude is maximal in this tetrahedron. Then we split the tetrahedron locally to incorporate the vertex. After several iterations of such split operation, the final initial tetrahedralization is constructed.

5.3. Trivariate DMS-spline Fitting with Free Knots

If only the control vectors are treated as variables in Equation 10, it falls into a very special category of nonlinear programming, i.e., unconstrained convex quadratic programming.

$$E = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} + f,$$

where $\mathbf{x} = (\dots, \mathbf{c}_\beta^I, \dots)^T$,

$$
Q = \left(
\begin{array}{ccc}
 & \vdots & \\
\dots & 2\sum_{i=1}^m N_\beta^I(x_i, y_i, z_i) N_{\beta'}^{I'}(x_i, y_i, z_i) & \dots \\
 & \vdots &
\end{array}
\right),
$$

$$\mathbf{c} = (\dots, -2\sum_{i=1}^m \mathbf{p}_i N_\beta^I(x_i, y_i, z_i), \dots)^T,$$

and $f = \sum_{i=1}^m \|\mathbf{p}_i\|^2$.

Note that, Q is a positive definite, symmetric and sparse matrix. Interior-point method can solve this problem very efficiently.

When considering the knots as free variables in Equation 10, we need calculate the gradient with respect to knots. We derive the formula of the directional derivative of trivariate DMS-spline $\mathbf{s}(\mathbf{u})$ with respect to a knot $\mathbf{t}_{p,l}$ ($p \in N$, $0 \leq l < n$) along the direction $\mathbf{v}$:

$$\mathcal{D}_{\mathbf{t}_{p,l}, \mathbf{v}} \mathbf{s}(\mathbf{u}) = \mathcal{D}_{\mathbf{v}} G(\mathbf{u}) + H(\mathbf{u}, \mathbf{v}), \qquad (11)$$

where

$$G(\mathbf{u}) = -\frac{1}{n+1} \sum_{I \in \Omega, i_j = p} \sum_{|\beta| = n+1, \beta_j \geq l} \mathbf{c}_{\beta - e^j}^I N(\mathbf{u}|\widehat{V}_\beta^I),$$

$$H(\mathbf{u}, \mathbf{v}) = \sum_{I \in \Omega, i_j = p} \sum_{|\beta| = n, \beta_j = l} \mu_j(\mathbf{v}|X_\beta^I) \mathbf{c}_\beta^I N(\mathbf{u}|V_\beta^I),$$

and

$$\widehat{V}_\beta^I = \{\dots, \mathbf{t}_{p,0}, \dots, \mathbf{t}_{p,l-1}, \mathbf{t}_{p,l}, \mathbf{t}_{p,l}, \mathbf{t}_{p,l+1}, \dots, \mathbf{t}_{p,n}, \dots, \}.$$

Note that the i_j in above equation is the j-th element of 4-tuple $I = (i_0, \dots, i_3)$ which represents the vertex indices of a tetrahedron I.

We also need to pay special attention to the positions of knots.

To describe clearly, we classify the knots into two categories: the primary knots $\{\mathbf{t}_{s,0}|s \in \mathbb{N}\}$ and the sub-knots $\{\mathbf{t}_{s,l}|s \in \mathbb{N}, 1 \leq l \leq n\}$.

The primary knots must yield valid tetrahedralization in Ω and the sub-knots must satisfy Equation 3. Especially, the sub-knots on the boundary must lie outside of Ω. To prevent degeneracy, we also set the restriction to the minimal length between any two knots (either primary knots or sub-knots). Therefore, Equation 10 is a typical large-scale constrained nonlinear programming problem. It is usually very time consuming for solve this kind of problem. To simplify our implementation and improve the performance, we solve this problem "locally", i.e., for each node $\mathbf{t}_i$ of the tetrahedralization, if the fitting error in its 1-ring neighboring tetrahedra is larger than a user-specified value, then we solve Equation 10 by treating the knots associated to $\mathbf{t}_i$ as free variables. Since all other knots are fixed, we can deal with a sub-problem of Equation 10, in which only

$$\{\mathbf{p}_j|\mathbf{p}_j \text{ is in 1-ring neighboring of } \mathbf{t}_i\}$$

are considered.

5.4. Local Adaptive Refinement

The above volume data fitting procedure attempts to minimize the total squared distance of the volume data points $\mathbf{p}_i$ to the DMS-spline $\mathbf{s}$. For some regions with dense points or sharp features, it is often desirable to introduce new degrees of freedom into the spline representation in order to improve the fitting quality. The goal of this refinement is to subdivide any domain tetrahedron whose fitting error is greater than a threshold. Essentially, this step is the knot insertion for DMS-spline. Seidel et al. [SV94] proved that the new control vectors of the refined domain can be computed with the polar form. To simplify our implementation, in this paper we simply treat all the control vectors as variables and solve Equation 10 again.

Intuitively, local adaptive refinement is a further action for allocating tetrahedra around the feature parts. Error characterization and evaluation is an important issue at this step. In the adaptive refinement, both geometric features and field features are considered.

For attribute data fitting, if the error metric inside a tetrahedron is greater than a user-specified value, a new knot is inserted at the location, where the gradient magnitude is largest inside the tetrahedron. For solid geometry fitting, a new inserted knot should be placed on the feature line. During the optimization, the primary knots can only move along the sharp feature. This is explicitly enforced since the feature lines have been detected. The sub-knots must lie on the feature line segment between two adjacent primary knots.

Figure 10 shows the fitting results for the spx model. Figure 10(a) and (b) shows the fitting with control vectors only, while Figure 10(c) and (d) shows the final fitting results with both control vectors and knots. Apparently, adjusting knots can greatly reduce the fitting error and achieve a better effect.

Figure 11 shows the fitting results for a smooth engineering part, router. Figure 11(a) is the original data set. Figure 11(b) shows the final fitting result, which is fitted using both control vectors and knots.

Figure 12 shows the fitting results for two separated engineering parts. Figure 12(a) shows the original data set. Figure 12(b) shows the final fitting result, which is fitted with both control vectors and knots.
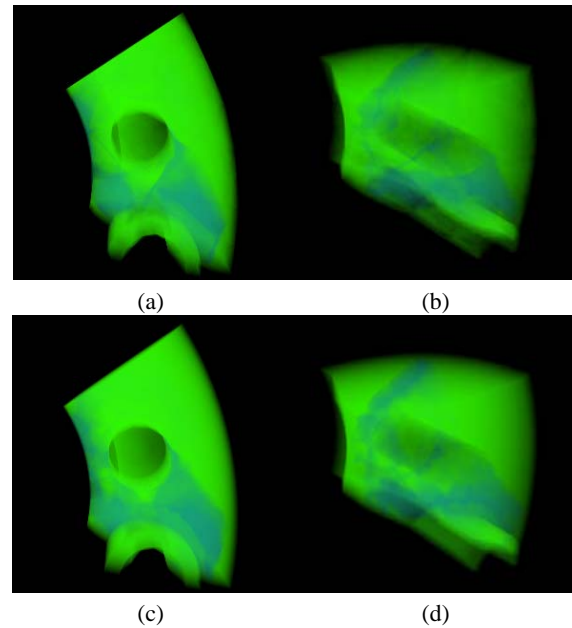

(a)   (b)

(c)   (d)

Figure 10: *(a)-(b) Fitting with control vectors only (front view and side view). (c)-(d) Fitting with both control vectors and knots (front view and side view).*
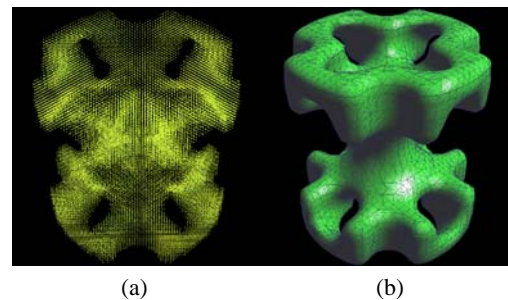

(a)   (b)

Figure 11: *(a) The original dataset, router, in point view, where the color indicates the attribute value. (b) Fitting with both control vectors and knots. The final result is rendered using our marching tetrahedra algorithm.*

6. Visualization Techniques

6.1. Direct Volume Rendering

Attribute distribution on a 3D solid object can be visualized in a number of ways, for example by color contours on a 2D slice, or by a polygonal approximation to a contour surface. Direct volume rendering refers to techniques which produce a projected image directly from the volume data, without intermediate constructs such as contour surface polygons. Among the direct volume rendering techniques, the ray casting approach typically casts rays from pixels in the screen into a volumetric dataset. The quantity that is accumulated for each ray originating in each pixel is converted into color or intensity and is assigned to the pixel. It assumes datasets scatter, occlude, generate, and reflect light. Volume rendering enhances three dimensional visualization of imaged tissue by providing translucent rendering. In addition to the standard 3D image analysis tools, vol-
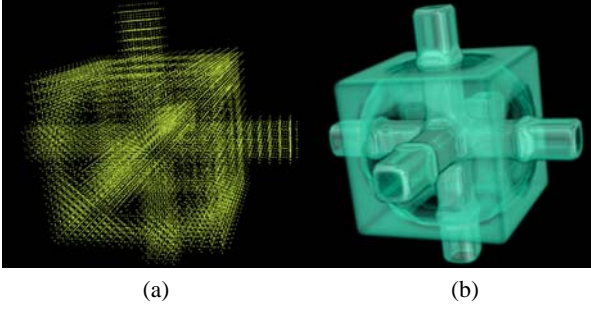
(a)   (b)

Figure 12: *(a) The original dataset, crosscube, in point view, where the color indicates the attribute value. (b) Fitting with both control vectors and knots. The final result is rendered using the direct volume rendering algorithm.*

ume rendering allows the user to interactively define thresholds for opacity, color application, and brightness. Translucent rendering of volumetric data provides more information about a spatial relationship of different structures than standard 3D surface rendering. Direct volume rendering affords to quickly isolate tissue of interest, and quickly provides 3D spatial information for enhanced diagnostic confidence, improves surgical and treatment planning, and aids in education. In the proposed approach, we only consider absorption nature of the dataset. That is to say that the object is visualized by integrating the density of the trivariate simplex functions along the path of each casted ray as in [RE01]. The intensity of light passing through translucent material decreases exponentially. Therefore,

$$I(a,b) \ = \ \exp(-\int_0^L \sum_{I \in \Omega} \sum_{|\beta|=n} g_\beta^I N_\beta^I(\mathbf{u}(t))dt)$$

$$= \ \exp(-\sum_{I \in \Omega} \sum_{|\beta|=n} g_\beta^I \int_0^L N_\beta^I(\mathbf{u}(t))dt).$$

The proposed direct rendering of scalar simplex spline functions is able to incrementally update complex volumetric data sets at interactive rates of several frames per second. Assume a control coefficient g_β^I changes with Δg_β^I. Then the new intensity of the pixel will be:

$$I^{new}(a,b) = I^{old}(a,b) \exp(-\Delta g_\beta^I \int_0^L N_\beta^I(\mathbf{u}(t))dt). \quad (12)$$

Since $N_\beta^I(\mathbf{u}(t)) = |d(\mathbf{p}_i,\mathbf{q}_j,\mathbf{r}_k,\mathbf{s}_l)|M(\mathbf{u}(t)|\mathbf{V}_\beta^I)$ as shown in Equation 5, the problem becomes how to evaluate $\int_0^L M(\mathbf{u}(t)|\mathbf{V}_\beta^I)dt$ efficiently. Grounded on the theory of simplex splines, we derive the following analytic solution to compute the integral in a recursive fashion. In the following derivation, the notations has the same meanings as in 3.1. In order to save space, we abbreviate $\mathbf{V}_\beta^I$ to $\mathbf{V}$. Suppose that $\mathbf{u}(t) = \mathbf{x}_c + t\mathbf{d}_c$, where $\mathbf{x}_c$ and $\mathbf{d}_c$ are constant vectors, which denote the starting point of a casted ray and the unit direction of the ray, respectively. Then when $n > 0$, using the linear decomposition, we can obtain:

$$\int_0^L M(\mathbf{u}(t)|\mathbf{V})dt = \int_0^L \sum_{j=0}^3 \lambda(\mathbf{x}_c + t\mathbf{d}_c)M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V}\setminus\{\mathbf{x}_{k_j}\})dt$$

$$= \int_0^L (\sum_{j=0}^3 \lambda(\mathbf{x}_c) + t\sum_{j=0}^3 \mu(\mathbf{d}_c))M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V}\setminus\{\mathbf{x}_{k_j}\})dt, \quad (13)$$

where λ and μ have the same meaning as in Equation 1 and 2. Due to the existing of Equation 2 and

$$\mathcal{D}_{\mathbf{d}_c}M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V}) = \mathbf{d}_c^\top \nabla M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V}) = \frac{dM(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})}{dt}, \quad (14)$$

then performing integration by parts, we can obtain,

$$\int_0^L t\sum_{j=0}^3 \mu(\mathbf{d}_c)M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V}\setminus\{\mathbf{x}_{k_j}\})dt$$

$$= \frac{1}{n}\int_0^L t\mathcal{D}_{\mathbf{d}_c}M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})dt$$

$$= tM(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})|_0^L - \frac{1}{n}\int_0^L M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})dt$$

$$= -\frac{1}{n}\int_0^L M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})dt. \quad (15)$$

Replacing Equation 15 in Equation 13, we can obtain the following equation:

$$\int_0^L M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})dt = \frac{n}{n+1}\sum_{j=0}^3 \lambda(\mathbf{x}_c)\int_0^L M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V}\setminus\{\mathbf{x}_{k_j}\})dt. \quad (16)$$

When $n = 0$,

$$\int_0^L M(\mathbf{x}_c + t\mathbf{d}_c|\mathbf{V})dt = \frac{L}{\text{Vol}(\mathbf{V})}.$$

With the above equation, we can efficiently evaluate the integral of densities along a casted line. Besides the X-ray volume rendering, we can also perform general ray casting. Since the solid object is represented by a single trivariate DMS-spline, our ray casting avoids resampling and interpolation problems.

6.2. Marching Tetrahedra Isosurfacing

Since the resulting heterogeneous solid object consists of tetrahedra, it is easy to perform marching tetrahedra isosurfacing to extract an isosurface of its associated field. Please refer to [CDM*94] for the detail of a marching tetrahedra algorithm. It is important to note that, based on the geometry of the primary knots, the samples may be spaced at non-unit intervals or distributed very irregularly. Our approach does not require resampling onto a regular voxel raster, which would introduce error. Furthermore, along tetrahedron edges, the process of isosurface extraction employs the original trivariate DMS-spline interpolation, which has the potential to avoid introducing sampling artifacts. Since traditional voxel-based sculpting systems employ trilinear interpolation, they exhibit aliasing when the voxel grid is scaled or deformed. Aliasing must then be eliminated by filtering most or all of the voxel grid. We avoid this problem by using what is essentially a higher-order C^{n-1} trivariate spline. If the objects modeled by our approach are originally smooth, they will remain smooth even if the control lattice is arbitrarily scaled or even deformed. If they contain C^0 or discontinuity region, these features will be preserved as well.

A key aspect of isosurface extraction is normal vector computation. With our polynomial-based approach, it can be determined

Model	data pts	domain tetra	control pts	knots	geometric fitting error	field fitting error
spx	15832	1914	3335	992	8.21121×10^{-7}	7.78170×10^{-4}
cubecross	24128	2245	3294	858	1.25994×10^{-6}	1.97281×10^{-4}
router	34744	2812	4239	1236	1.13400×10^{-6}	1.52832×10^{-4}

Table 1: *Statistics of Data fitting.*

analytically. Furthermore, the local evaluation can lead to multiresolution isosurface extraction. When a tetrahedron of the object is determined cross the isosurface of its associated field, and its size is larger than a specified threshold, we can evaluate the trivariate DMS-spline and upsample it locally to increase the resolution and then perform isosurfacing on those smaller cells. Please see the illustration in Figure 13.
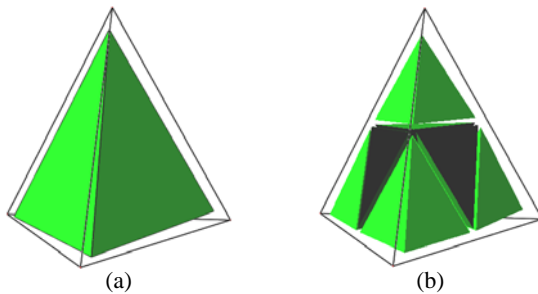
Figure 13: *(a) The single tetrahedron in the solid object is larger than the specified size and is needed to be upsampled. (b) Based on the corresponding domain tetrahedron (as shown in mesh), the tetrahedron in the solid object is upsampled. Each cell is scaled a little in order to clearly observe the upsampling result.*

7. Implementation and Discussion

We have implemented a prototype system on a PC with 3.0 GHz P4 CPU and 2GB of RAM. The system is written in C++ and OpenGL. In our system we incorporate the Phantom 1.0 device from Sensable Technologies into our user interface. Users can use a mouse or the Phantom to manipulate the control points or sketch the skeletons.

Table 1 shows the statistics of the performance of our fitting algorithm on several datasets, where the fitting error is the mean square error.

As one can see, we couple geometric and attribute representation together in order to provide a unified paradigm to explicitly model geometry, topology, and its associated attribute properties. However, one weak point of this representation is that if the geometric features and attribute field features are not conforming to each other, we have to select the higher resolution to model both, even though it might not necessary for one of them to have such a high resolution. A potential solution for this case is to decouple geometric and attribute representations. Thus, it offers more flexibility to construct different resolution of tetrahedralization of a domain to fit the geometry and attribute distribution, respectively. But special care of aligning the boundaries of two domains exactly the same must be taken, or

else some locations of the geometric object may not have attribute properties, or attribute properties are assigned to the locations outside the geometric object. When using the decoupling scheme, it is possible to model any number of attribute properties over the geometric object. Complicated geometries with simple attributes, and vice versa, may be represented at the resolution that best suits them. The gain may be a large savings in storage and execution time.

Furthermore, noise is an important variable in any visualization involving measured data. Simplex splines generally provide a robust representation for a signal containing moderate noise. Compared with polygons, or higher dimensional analogues such as voxels, simplex splines generally represent a smooth function with fewer points. However, with our knots manipulation technique they can represent higher frequency features as well.

8. Conclusion

In this paper we have articulated a new integral approach for representing, modeling, reconstructing, and visualizing multidimensional, physical attributes across any volumetric objects. In particular, we employ a trivariate simplex spline model that is defined over a tetrahedral decomposition of any 3D domain. The modeled volume can be of complicated geometry and arbitrary topology. The trivariate simplex spline model ideally serves the pressing needs of both visualization and modeling communities since the model makes use of a more general and flexible tetrahedral domain and offers a compact continuous representation at the same time. More importantly, it integrates geometry and attribute properties over domains of arbitrary topology, by modeling a complicated heterogeneous object with a single trivariate simplex spline without any additional operations of trimming or patching. All of the above attractive characteristics result from many appealing properties of trivariate simplex splines such as piecewise polynomials of low degree, high-order continuity, and sharp feature modeling through different knot placements. In addition, we have developed a feature-sensitive fitting algorithm that can reconstruct a more compact trivariate simplex spline from a structured or unstructured volume. It reconstructs the geometry and the associated material attributes simultaneously, satisfying various continuity requirements set by the user. Such flexibility allows us to model continuous or discontinuous material distribution with ease.

Our new representation can also facilitate multiresolution, local adaptive subdivision, free-form deformation, both isosurface and volume rendering, as well as other modeling and visualization functionalities. This is mainly because many of its nice analytical properties and associated computational tools. Based on our extensive experiments on using the trivariate simplex spline, we believe that our new paradigm can significantly advance the current state of the

knowledge in modeling and visualizing heterogeneous models of physical objects and their material properties. Besides solid modeling and volume visualization, in the near future we are pursuing other application directions including general data modeling, material editing and reconstruction, geometric processing of bio-medical and geological datasets, dynamic simulation and analysis of multi-resolution, heterogeneous objects, etc.

References

[CCM*00] CIGNONI P., CONSTANZA D., MONTANI C., ROCCHINI C., SCOPIGNO R.: Simplification of tetrahedral meshes with accurate error evaluation. In *Proceedings of the conference on Visualization '00* (2000), pp. 85–92.

[CDM*94] CIGNONI P., DE FLORIANI L., MONTANI C., PUPPO E., SCOPIGNO R.: Multiresolution modeling and visualization of volume data based on simplicial complexes. In *Proceedings of the 1994 symposium on Volume visualization* (1994), pp. 19–26.

[dB76] DE BOOR C.: Splines as linear combinations of B-splines. In *Approximation Theory II* (1976), Academic Press, New York, pp. 1–47.

[DMS92] DAHMEN W., MICCHELLI C. A., SEIDEL H.-P.: Blossoming begets B-spline bases buit better by b-patches. *Mathematics of Computation 59*, 199 (1992), 97–115.

[ES92] EDELSBRUNNER H., SHAH N. R.: Incremental topological flipping works for regular triangulations. In *Proceedings of 8th Annual ACM Symposium on Computational Geometry* (1992), pp. 43–52.

[GS94] GREINER G., SEIDEL H. P.: Modeling with triangular B-splines. *IEEE Computer Graphics and Applications 14*, 2 (1994), 56–60.

[HK03] HONG W., KAUFMAN A.: Feature preserved volume simplification. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications* (2003), pp. 328–333.

[HQ01] HUA J., QIN H.: Haptic sculpting of volumetric implicit functions. In *Proceedings of 9th Pacific Conference on Computer Graphics and Applications* (2001), pp. 254–264.

[HQ02] HUA J., QIN H.: Haptics-based volumetric modeling using dynamic spline-based implicit functions. In *Proceedings of IEEE Symposium on Volume Visualization and Graphics* (2002), pp. 55–64.

[HQ04] HE Y., QIN H.: Surface reconstruction with triangular B-splines. In *Proceedings of Geometric Modeling and Processing,* (2004).

[Joe91] JOE B.: Construction of three-dimensional triangulations using local transformations. *Computer Aided Geometric Design 8* (1991), 123–142.

[LJ94] LIU A., JOE B.: Relationship between tetrahedron shape measures. *BIT 34*, 2 (1994), 268–287.

[MC01] MARTIN W., COHEN E.: Representation and extraction of volumetric attributes using trivariate splines: A mathematical framework. In *Proceedings of the 7th ACM Symposium on Solid Modeling and Applications* (2001), pp. 234–240.

[PS94] PFEIFLE R., SEIDEL H.-P.: Faster evaluation of quadratic bivariate dms spline surfaces. In *Proceedings of Graphics Interface '94* (1994), pp. 182–189.

[PS95] PFEIFLE R., SEIDEL H.-P.: Fitting triangular B-splines to functional scattered data. In *Proceedings of Graphics Interface '95* (1995), pp. 80–88.

[QT96] QIN H., TERZOPOULOS D.: D-NURBS: A physics-based framework for geometric design. *IEEE Trans. on Visualization and Computer Graphics 2*, 1 (1996), 85–96.

[QT97] QIN H., TERZOPOULOS D.: Triangular NURBS and their dynamic generalizations. *Computer Aided Geometric Design 14*, 4 (1997), 325–347.

[RE99] RAVIV A., ELBER G.: Three dimensioinal freeform sculpting via zero sets of scalar trivariate functions. In *Proceedings of 5th ACM Symposium on Solid Modeling and Applications* (1999), pp. 246–257.

[RE01] RAVIV A., ELBER G.: Interactive direct rendering of trivariate B-spline scalar functions. *IEEE Transactions on Visualization and Computer Graphics 7*, 2 (2001), 109–119.

[RN00] ROXBOROUGH T., NIELSON G. M.: Tetrahedron based, least squares, progressive volume models with application to freehand ultrasound data. In *Proceedings of the conference on Visualization '00* (2000), pp. 93–100.

[She97] SHEWCHUK J. R.: *Delaunay Refinement Mesh Generation*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1997.

[SPS01] SCHMITT B., PASKO A., SCHLICK C.: Constructive modeling of frep solids using spline volumes. In *Proceedings of the 6th ACM symposium on Solid modeling and applications* (2001), pp. 321–322.

[SV94] SEIDEL H.-P., VERMEULEN A. H.: Simplex splines support surprisingly strong symmetric structures and subdivision. In *Curves and Surfaces II*. AK Peters, 1994, pp. 443–455.

[THJ99] TROTTS I. J., HAMANN B., JOY K. I.: Simplification of tetrahedral meshes with error bounds. *IEEE Trans. on Visualization and Computer Graphics 5*, 3 (1999), 224–237.

[YT00] YAO J., TAYLOR R. H.: Tetrahedral mesh modeling of density data for anatomical atlases and intensity-based registration. In *Proceedings of the third International Conference on Medical Image Computing and Computer-Assisted Intervention* (2000), pp. 531–540.