Springer

Journal of Mechanical Science and Technology 23 (2009) 1967~1981

Journal of
Mechanical
Science and
Technology

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-009-0513-2

A rational B-spline hypervolume for multidimensional multivariate modeling[†]

Sangkun Park[*]

Department of Mechanical Engineering, Chungju National University, 72 Daehangno, Chungju-si, 380-702, Korea

(Manuscript Received April 7, 2009; Revised May 7, 2009; Accepted May 11, 2009)

Abstract

This paper proposes a rational B-spline hypervolume that represents a volume object which has multiple attributes defined in a multidimensional space. This representation provides a mathematical framework for modeling and visualizing a multidimensional multivariate object as well as analyzing the object interiors to extract its intrinsic features that are directly inaccessible. We discuss the NURBS extension procedure showing that the proposed hypervolume is a generalized volume function not depending on the domain dimensionality and its range dimensionality. Useful expressions arising in connection with a computational treatment are presented for geometric and mathematical analysis of a volume object based on the proposed hypervolume. We also describe the approximation and interpolation algorithms of the proposed hypervolume. Finally, we show various applications such as grid generation, flow visualization, implicit surface modeling, and image morphing. They demonstrate the usefulness and the extensibility of the proposed hypervolume.

Keywords: Hypervolume; Multidimensional multivariate; Rational B-spline; Scattered data approximation; Volume modeling

1. Introduction

Many research groups in various areas have made efforts to model and analyze very large data sets efficiently as well as to visualize the results effectively. Such efforts [1-4] include topics in flow visualization [5] in computational fluid dynamics, volume graphics in computer graphics, and scattered data modeling [6-9] in CAGD community.

These research efforts mainly aim to devise mathematical principles, computational algorithms, and well-organized data structures, which transform massive volume data sets into meaningful pictures and other graphic representations that improve apprehension or inspiration. To achieve these goals, many modeling techniques and visualization tools have been integrated to some degree within a system. Few systems, however, provide a unified working environment by which users can perform the modeling (or representation), analysis, and visualization of various types of volume data (e.g., heterogeneous materials, CT images, flow visualization). In other words, system developers have difficulties in constructing a unified framework (independent of data complexities and domain dimensionality) that can support all types of volume models and various tasks. These difficulties arise basically from the lack of generalized mathematical models that represent a wide range of volume data and carry out the existing volume visualization or analysis techniques with their least modification. Needs for the generalized models can be found even in the historical evolution of volume models (refer to section 2). The evolving trend of volume object or models can be summarized as:

- From discrete volume datasets to continuous func-

[†] This paper was recommended for publication in revised form by Associate Editor Dae-Eun Kim

[*] Corresponding author. Tel.: +82 43 841 5122, Fax.: +82 43 841 5120
E-mail address: skpark@cjnu.ac.kr

© KSME & Springer 2009

tion models

- From homogeneous solids to heterogeneous volumes
- From Boolean domain to real domain
- From an one scalar field to a generic tuple of scalar fields
- From three-dimensional space to higher dimensional space
- From a single representation to a constructive modeling approach for complex objects

From the facts above, the domain complexity and range complexity of a volume object need to be considered for multidisciplinary volume objects in various applications.

In general, a volume object is defined as a function $V : S \rightarrow A$ from a three-dimensional compact domain S to an attribute space A. That is, a volume object is composed of an infinite number of point particles in three-dimensional Euclidean space where each of the particles has multiple attributes as object properties depending on application purposes. Typical examples of attributes include geometric attributes (e.g., algebraic distance, rectilinear coordinates), physical field attributes (e.g., density, velocity, temperature, pressure), photometric attributes (e.g., color, opacity), and other application-dependent attributes. Here, the geometric attribute is often described by using the rectilinear coordinates (x, y, z) in 3D Euclidean space as an essential element of the volume object, while the other attributes are mostly represented by real-valued scalar functions to define the intrinsic properties of the object varying across a region of the space.

On the other hand, three-dimensional domain space of a volume object can be extended to so-called multidimensional space in which a multivariate volume object is described. Note that the term *multidimensional* refers to the domain dimensionality of a volume space, while the term *multivariate* refers to the range dimensionality of volume attributes. Therefore, a generalized volume model has to represent a multidimensional multivariate (in short, *mdmv*) volume object, called a hypervolume object in this paper. A typical example is discrete scalar fields embedded in D-dimensional spaces, which is discussed by Bajaj et al. [10]. The research topic of the *mdmv* was among one of the short-term goals included in the 1987 National Science Foundation (NSF) sponsored workshop on *Visualization in Scientific Computing* [11]. Many attempts for effective and efficient *mdmv* visualization techniques have been tried since then. However, there has been little work on developing a generalized volume model that fully represents a wide range of volume data sets without depending on a data dimensionality and distribution over a multidimensional domain space.

Hence, we introduce a rational B-spline hypervolume which represent a generalized volume object having K attributes in D-dimensional Euclidean space. It is a generalized form of NURBS [12] representation that has become the de facto industry standard for curve and surface description in the CAD community. The generalization process is carried out by extending the parametric space of NURBS to an arbitrary D-dimensional space for multidimensional domain as well as the vector size of NURBS control point to an arbitrary number of K for multivariate attributes. That is, the volume representation model presented in this paper is the extension version of NURBS curves or surfaces to multivariate volume objects defined in multidimensional space. We expect this model to establish a mathematical framework that provides a foundation for representing and visualizing a general volume object.

2. Related work

Research in modeling a multidimensional multivariate object has been studied in various application areas. We present a brief survey here of volume models that have contributed to the volume modeling community on the order of complexity of a volume object.

A discrete field volume is a sort of a finite set of volume elements each of which has a 3D spatial location and a numerical data value, distributed over a bounded region of space. It has a topology and geometry to describe the connections between its elements. An advantage of the discrete volume is that it can represent heterogeneous volumes as well as regular solids. It is also relatively easy to directly render these types of objects [13, 14] using discrete raycasting. A disadvantage is the larger amount of memory overhead normally associated with this representation.

A blobby object has been introduced to model and render electron clouds that surround their constituent atoms. Blinn [15] used exponentially decaying Gaussian functions created by each atom, and defined the iso-surfaces for the identical electron densities by summing the contributions from all fields of influence generated by each density function.

An implicit surface represents a surface as the zero set of a real-valued scalar function $f(x, y, z) = 0$. In practice, an implicit surface or function is mathematically defined such that it has first-degree continuity. However, it is difficult to render them in their original form, and thus they are usually polygonized by using [16], for example, Marching Cubes [17].

Function representation (F-rep) is concerned with the geometric objects represented by closed subsets of D-dimensional Euclidean space with the definition $f(x_1, x_2, \cdots, x_D) \geq 0$ where f is a real-valued continuous function. The F-rep [18] modeling components include the sets of objects, operations and relations. Algebraic surfaces, skeleton-based convolution surfaces, swept objects, volume dataset, and procedural models (such as solid noise) can be used as the primitive object. The set-theoretic operations, blending, offsetting, and other non-linear deformation are included in the modeling operations. And the relations describe the inclusion (binary), the point membership (in/boundary/out), and the intersection (collision) information.

Complex objects are often made by recursively applying constructive operations and transformations to simpler ones. These building processes are represented as a hierarchical constructive tree that includes the non-terminal nodes for operators applied to their children and the terminal nodes for defining primitives. Several major frameworks for building constructive representations of volume objects have been proposed: constructive solid geometry (CSG) [19], volumetric CSG [20], constructive volume geometry (CVG) [21], and constructive hypervolume modeling [22].

3. Rational B-spline hypervolumes

3.1 Volume objects

We first generally describe a volume object mentioned in many research groups before describing a rational B-spline hypervolume proposed in this paper.

Let $\mathbf{R}$ denote the set of all real numbers, and $\mathbf{E}^D$ denote D-dimensional Euclidean space. A general volume object $\mathbf{o}$ (often called infinite spatial object) can be expressed as a generic tuple of K (> 0) scalar fields, f_i $(i = 1, \cdots K)$ where $f_i : \mathbf{E}^D \rightarrow \mathbf{R}$. In other words, it is written as:

$$\mathbf{o} = \left(f_1, f_2, \cdots, f_K\right) \tag{1}$$

where f_i is a real-valued function representing an i-th attribute of a volume object distributed over D-dimensional space. This volume object must be bounded if it is to be made practical in many application areas. For this purpose, we define the bounded volume object as follows:

$$\mathbf{o} = \left(f_1(\mathbf{p}), f_2(\mathbf{p}), \cdots, f_K(\mathbf{p})\right), \ \mathbf{p} \in \mathbf{\Omega}^D, \ \mathbf{\Omega}^D \subset \mathbf{E}^D \tag{2}$$

where $\mathbf{p}$ and $\mathbf{\Omega}^D$ denote a D-dimensional point and a point set defined on the bounded space in $\mathbf{E}^D$.

To facilitate volume modeling and rendering, the bounded point set is described by a parametric unit space and a space transformation function, $\mathbf{T}$. Here, $\mathbf{T}$ is a mapping function from the parametric unit space to the original region $\mathbf{\Omega}^D$. That is,

$$\mathbf{p} = \mathbf{T}(\mathbf{u}), \ \mathbf{u} \in [0,1]^D \tag{3}$$

where $\mathbf{u}$ is a point defined in the parametric unit space. Assuming $D = 3$, the space transformation from Cartesian coordinates to cylindrical coordinates can be an example of $\mathbf{T}$.

The rational B-spline hypervolume proposed in this paper satisfies Eqs. (2) and (3) for a general volume object. That is, it contains a generic tuple of K scalar fields, a parametric domain defined in D-dimensional space, and a mapping function for space transformation. The term "control point" of the rational B-spline hypervolume is used to describe K attribute fields and the term "knot vectors" corresponds to the D-dimensional parametric domain. In fact, the product of "control point" and "B-spline function" behaves as an attribute function that combines a real-valued function f with a space transformation $\mathbf{T}$ (i.e., $f(\mathbf{p}) = f(\mathbf{T}(\mathbf{u}))$).

3.2 Definition and properties

This section shows that the proposed rational B-spline hypervolume is a generalized volume model function that represents a multivariate volume object embedded in multidimensional space. We first describe a simple parametric function defined in one-dimensional space with an attribute and then extend it to the D-dimensional space with K attribute fields. A simple parametric function is defined as follows by the non-uniform B-spline function of order k in the u

direction.

$$A(u) = \sum_{i=0}^{n-1} A_i N_i^k(u) \tag{4}$$

where $A_i \subset \mathbf{R}^1$ is the i-th control value, n is the number of control values, and $N_i^k(u)$ is the normalized B-spline basis function defined on the knot vector, $\mathbf{U} = \{t_i\}_{i=0}^{i=n+k-1}$.

This parametric function is now extended to a generalized volume function by two steps. One is to extend the function to have K attributes, and the other is the extension to a D-dimensional space.

- Extension to multiple attributes

The extension of Eq. (4) for multiple attributes is easily made by inserting a new control value to the control vector. Thus, the one dimensional parametric function with K attribute fields is written in the vector form like:

$$\mathbf{A}(u) = \sum \mathbf{A}_i N_i(u) \tag{5}$$

where $\mathbf{A}(u) = \begin{pmatrix} A^1(u) \\ A^2(u) \\ \vdots \\ A^K(u) \end{pmatrix}$ and $\mathbf{A}_i = \begin{pmatrix} A^1 \\ A^2 \\ \vdots \\ A^K \end{pmatrix}_i$.

Note that $A^j(u)$ $(j = 1, \cdots, K)$ represents the j-th attribute function among K attribute functions, and $(A^j)_i$ describes the j-th attribute of the i-th control point of n control points.

- Extension to multidimensional space

The extension of Eq. (5) for multidimensional space is made by using tensor product schemes [23]. The extension steps for r and (r+1)-dimensional space (r = 1, ⋯ , D-1) are written as follows:

Step (r) :

$$\mathbf{A}(u_1, \cdots, u_r) = \sum_{i_1} \cdots \sum_{i_r} \mathbf{A}_{i_1 \cdots i_r} N_{i_r}(u_r) \cdots N_{i_1}(u_1) \tag{6}$$

Step (r+1) :

$$\begin{aligned} &\mathbf{A}(u_1, \cdots, u_r, u_{r+1}) \\ &= \sum_{i_1} \cdots \sum_{i_r} \mathbf{A}_{i_1 \cdots i_r}(u_{r+1}) N_{i_r}(u_r) \cdots N_{i_1}(u_1) \\ &= \sum_{i_1} \cdots \sum_{i_{r+1}} \mathbf{A}_{i_1 \cdots i_{r+1}} N_{i_{r+1}}(u_{r+1}) \cdots N_{i_1}(u_1) \end{aligned} \tag{7}$$

Thus, the multidimensional function with multiple attributes is written in the following vector form:

$$\mathbf{A}(\mathbf{u}) = \sum_{\mathbf{I}} \mathbf{A}_{\mathbf{I}} \mathbf{N}_{\mathbf{I}}(\mathbf{u}) \tag{8}$$

where $\mathbf{A}(\mathbf{u}) = \mathbf{A}(u_1, \cdots, u_D)$, $\sum_{\mathbf{I}} = \sum_{i_1} \cdots \sum_{i_D}$, $\mathbf{A}_{\mathbf{I}} = \mathbf{A}_{i_1 \cdots i_D}$, and $\mathbf{N}_{\mathbf{I}}(\mathbf{u}) = N_{i_1}(u_1) \cdots N_{i_D}(u_D)$.

In addition, Eq. (8) can be extended to a rational form as follows:

$$\begin{aligned} \mathbf{A}(\mathbf{u}) &= \frac{\sum_{\mathbf{I}} h_{\mathbf{I}} \mathbf{A}_{\mathbf{I}} \mathbf{N}_{\mathbf{I}}(\mathbf{u})}{\sum_{\mathbf{I}} h_{\mathbf{I}} \mathbf{N}_{\mathbf{I}}(\mathbf{u})} \\ &= \sum_{\mathbf{I}} \mathbf{A}_{\mathbf{I}} \frac{h_{\mathbf{I}} \mathbf{N}_{\mathbf{I}}(\mathbf{u})}{\sum_{\mathbf{J}} h_{\mathbf{J}} \mathbf{N}_{\mathbf{J}}(\mathbf{u})} \\ &= \sum_{\mathbf{I}} \mathbf{A}_{\mathbf{I}} \mathbf{R}_{\mathbf{I}}(\mathbf{u}) \end{aligned} \tag{9}$$

where $\mathbf{R}_{\mathbf{I}}(\mathbf{u}) = \dfrac{h_{\mathbf{I}} \mathbf{N}_{\mathbf{I}}(\mathbf{u})}{\sum_{\mathbf{J}} h_{\mathbf{J}} \mathbf{N}_{\mathbf{J}}(\mathbf{u})}$, $h_{\mathbf{I}}$ is called a homogeneous coordinate and often used as a weight of the **I**-th control vector (or point).

Therefore, we achieve a rational B-spline hypervolume that has multivariate attributes distributed over a bounded region of multidimensional parameter space, which is given by:

3.3 Definition of a rational B-spline hypervolume

$$\begin{aligned} \mathbf{A}(u_1, \cdots, u_D) &= \begin{pmatrix} A^1(u_1, \cdots, u_D) \\ \vdots \\ A^K(u_1, \cdots, u_D) \end{pmatrix} \\ &= \frac{\sum_{i_1=0}^{n_1-1} \cdots \sum_{i_D=0}^{n_D-1} \begin{pmatrix} hA^1 \\ \vdots \\ hA^K \end{pmatrix}_{i_1 \cdots i_D} N_{i_1}^{k_1}(u_1) \cdots N_{i_D}^{k_D}(u_D)}{\sum_{i_1=0}^{n_1-1} \cdots \sum_{i_D=0}^{n_D-1} h_{i_1 \cdots i_D} N_{i_1}^{k_1}(u_1) \cdots N_{i_D}^{k_D}(u_D)} \end{aligned} \tag{10}$$

where u_i $(i = 1, \cdots, D)$ is the i-th parameter to define a D-dimensional volume space and $A^j(u_1, \cdots, u_D)$ $(j = 1, \cdots, K)$ represents the j-th attribute function among K attribute functions. n_i and k_i are the number of control points and the order of the B-spline basis function along the u_i parametric direction, respectively. $(A^j)_{i_1 \cdots i_D}$ describes the j-th attribute of the (i_1, i_2, …, i_D)-th control point of $n_1 \times n_2 \times \cdots \times n_D$ control points. $h_{i_1 \cdots i_D}$ is a homogeneous coordinate and $N_{i_i}^{k_i}(u_i)$ denotes the normalized B-spline basis function of order k_i defined on the knot vector in the u_i direction. Knot vectors for D-dimensional parametric domain of Eq. (10) are defined as follows.

Knot vector in u_1 direction, $\mathbf{U}_1 = \left\{t_{i_1}^{(1)}\right\}_{i_1=0}^{i_1=n_1+k_1-1}$

………

Knot vector in u_D direction, $\mathbf{U}_D = \left\{t_{i_D}^{(D)}\right\}_{i_D=0}^{i_D=n_D+k_D-1}$

As a special case of Eq. (10), NURBS curves (or surfaces) used in CAD/CAM applications can be easily derived from Eq. (10). Applying D = 1, $\mathbf{u}$ = t, K = 3, and $\mathbf{A_I} = (x_i, y_i, z_i)$ to Eq. (10), the rational B-spline hypervolume is simplified into a NURBS curve as shown in Eq. (11).

$$\mathbf{A}_C(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \frac{\sum_i \begin{pmatrix} hx \\ hy \\ hz \end{pmatrix}_i N_i(t)}{\sum_i h_i N_i(t)} \tag{11}$$

Applying D = 2, $\mathbf{u}$ = (u, v), K = 3, and $\mathbf{A_I} = (x_{ij}, y_{ij}, z_{ij})$ to Eq. (10), we obtain a NURBS surface as shown in Eq. (12).

$$\mathbf{A}_S(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \frac{\sum_i \sum_j \begin{pmatrix} hx \\ hy \\ hz \end{pmatrix}_{ij} N_i(u) N_j(v)}{\sum_i \sum_j h_{ij} N_i(u) N_j(v)} \tag{12}$$

From this, we can achieve a NURBS volume by using D = 3, $\mathbf{u}$ = (u, v, w), K = 3, and $\mathbf{A_I} = (x_{ij}, y_{ij}, z_{ij})$. Furthermore, we are able to build a general purpose fluid flow visualization model [24], a NURBS volume with multiple flow attributes.

The proposed B-spline hypervolume has several advantageous properties as follows, which should be taken into account for a generalized volume representation:

- *Multidimensional objects*: It describes a volume object existing in multidimensional space. In other words, it defines the D-dimensional (D > 0) volume space where D denotes the domain dimensionality. The parameter space of a rational B-spline hypervolume corresponds to the volume space in which a volume object exists.
- *Multivariate objects*: It represents an arbitrary number of heterogeneous attributes distributed inside a volume object. That is, it contains K (K > 0) attributes where K indicates the range dimensionality. The control point of a rational B-spline hypervolume corresponds to the attributes of a volume object.
- *Independence of mathematical property*: It has no constraints regarding to the mathematical natures of attribute such as scalar, vector, and tensor, and is not dependent on the numerical types such as Boolean, integer, real, and so on. For example, assuming a volume model needs to represent temperatures, fluid velocities, and stresses at the same time, temperature (a scalar) can be represented by A^1, fluid velocity (a vector with three attributes) by A^2, A^3, A^4, and stress (a tensor with six attributes) by A^5 to A^{10} in Eq. (10).
- *A variety of applications*: It provides a mathematical foundation for geometric modeling, volume visualization, and engineering analysis conducted in many applications. For example, it enables us to compute various differential elements and derivative operators which are frequently required for geometric analysis and feature-based visualization. Application examples in section 5 show its capabilities to be used for various applications.
- *Inheritance from NURBS*: It has many useful properties studied in the previous NURBS literature [12, 25]: convex hull property, affine invariance, variation diminishing property, and local modification, etc. We can also make use of NURBS-based algorithms like degree reduction, degree elevation and knot insertion for a rational B-spline hypervolume without a special modification. Note that this inheritance is not related to a generalized volume. But this allows a fast and robust numerical computation, which has been already proved in NURBS literature.

4. Construction techniques by data fitting

4.1 Approximation of scattered data

In this section, we will introduce the approximation algorithm that constructs a rational B-spline hypervolume $\mathbf{A}(\mathbf{u})$ with L control points from the irregularly scattered data point set $\{\mathbf{P}_i\}$ of size M. The basic ideas of the algorithm come from least squares and pseudo-inverse (or Moor-Penrose generalized inverse) techniques. We assume that all knot vectors of a rational B-spline hypervolume are uniform and all homogeneous coordinates are 1.0. ($h_{\mathbf{I}} = 1.0$)

The approximation algorithm presented in this paper is classified into two categories: over-determined constraint problem and under-determined constraint problem. In the former case, the number of the given data points is greater than the number of the unknown control points $\mathbf{A_I}$ (i.e., $M > L$) and vice versa in the latter case. To derive the computation algorithm in each constraint problem, we select Eq. (13) as the proposed hypervolume model, which is a non-rational form for simple description.

$$\begin{aligned}&\mathbf{A}(u_1,\cdots,u_D)\\&=\sum_{i_1=0}^{n_1-1}\cdots\sum_{i_D=0}^{n_D-1}\mathbf{A}_{i_1\cdots i_D}N_{i_1}^{k_1}(u_1)\cdots N_{i_D}^{k_D}(u_D)\\&=\sum_{\mathbf{I}=0}^{L-1}\mathbf{A_I}\mathbf{N_I}(u_1,\cdots,u_D)\end{aligned}\tag{13}$$

where the number of control points $\mathbf{A_I}$ is $L = n_1 \times \cdots \times n_D$, and the index $\mathbf{I}$ of $\mathbf{A_I}$ is given by

$$\mathbf{I} = i_1 + (n_1)\cdot i_2 + (n_1 n_2)\cdot i_3 + \cdots + (n_1 \cdots n_{D-1})\cdot i_D$$

Both constraint problems can be formulated as one matrix form shown in Eq. (14). When the data points $\mathbf{P}_i$ $(i = 0,\cdots,M-1)$ and the corresponding parameter coordinates $(u_1^i,\cdots,u_D^i)$ $(i = 0,\cdots,M-1)$ are given and substituted into Eq. (13), the following matrix equation can be obtained after some manipulation.

$$\begin{Bmatrix}\vdots\\\mathbf{P}_i\\\vdots\end{Bmatrix}=\begin{bmatrix}\cdots & \mathbf{N}_{\mathbf{I}}^{0} & \cdots\\ & \vdots & \\ \cdots & \mathbf{N}_{\mathbf{I}}^{M-1} & \cdots\end{bmatrix}\begin{Bmatrix}\vdots\\\mathbf{A_I}\\\vdots\end{Bmatrix}\tag{14}$$

or

$$\{\mathbf{P}_i\} = \left[\mathbf{N}_{\mathbf{I}}^i\right]\{\mathbf{A_I}\} \quad (i = 0,\cdots,M-1)\ (\mathbf{I} = 0,\cdots,L-1)$$

where $\{\mathbf{P}_i\}$ represents the data point vector of size M and $\{\mathbf{A_I}\}$ denotes the control point vector of size L. Also, $\mathbf{N}_{\mathbf{I}}^i = \mathbf{N_I}(u_1^i,u_2^i,\cdots,u_D^i) = N_{i_1}^{k_1}(u_1^i)\cdots N_{i_D}^{k_D}(u_D^i)$ and $\left[\mathbf{N}_{\mathbf{I}}^i\right]$ is a $M \times L$ matrix.

4.1.1 Over-determined constraint problem

Applying least-squares technique to determine the unknown $\{\mathbf{A_I}\}$, we obtain the following Eq. (15) for over-determined constraint problem (i.e., $M > L$).

$$\left[\mathbf{N}_{\mathbf{I}}^i\right]^T\{\mathbf{P}_i\} = \left[\mathbf{N}_{\mathbf{I}}^i\right]^T\left[\mathbf{N}_{\mathbf{I}}^i\right]\{\mathbf{A_I}\}\tag{15}$$

After some manipulation of Eq. (15), we can get

$$\left\{\sum_{i=0}^{M-1}N_{\mathbf{I}}^i\mathbf{P}_i\right\} = \left[a_{\mathbf{IJ}} = \sum_{i=0}^{M-1}N_{\mathbf{I}}^i N_{\mathbf{J}}^i\right]\{\mathbf{A_J}\}\tag{16}$$

where $\mathbf{I} = 0,\cdots,L-1$, $\mathbf{J} = 0,\cdots,L-1$.

The matrix $\left[a_{\mathbf{IJ}}\right]$ shown in Eq. (16) is symmetric and positive semi-definite, so that we can obtain the unique solution of Eq. (16). In other words, the unknown control points $\{\mathbf{A_I}\}$ are uniquely determined.

4.1.2 Under-determined constraint problem

Applying the pseudo-inverse technique to determine the unknown $\{\mathbf{A_I}\}$, we get the following Eq. (17) for the under-determined constraint problem (i.e., $M < L$).

$$\left(\left[\mathbf{N}_{\mathbf{I}}^i\right]\left[\mathbf{N}_{\mathbf{I}}^i\right]^T\right)\left(\left[\mathbf{N}_{\mathbf{I}}^i\right]\left[\mathbf{N}_{\mathbf{I}}^i\right]^T\right)^{-1}\{\mathbf{P}_i\} = \left[\mathbf{N}_{\mathbf{I}}^i\right]\{\mathbf{A_I}\}\tag{17}$$

First, we obtain $\{\mathbf{Q}_j\}$ from $\left(\left[\mathbf{N}_{\mathbf{I}}^i\right]\left[\mathbf{N}_{\mathbf{I}}^i\right]^T\right)^{-1}\{\mathbf{P}_i\} = \{\mathbf{Q}_j\}$ where $\left[\mathbf{N}_{\mathbf{I}}^i\right]\left[\mathbf{N}_{\mathbf{I}}^i\right]^T$ is symmetric and positive semi-definite because it can be rewritten as Eq. (18).

$$\left[\mathbf{N}_{\mathbf{I}}^i\right]\left[\mathbf{N}_{\mathbf{I}}^i\right]^T = \left[a_{ij} = \sum_{\mathbf{I}=0}^{L-1}\mathbf{N}_{\mathbf{I}}^i\mathbf{N}_{\mathbf{I}}^j\right]\tag{18}$$

where $i = 0,\cdots,M-1$, $j = 0,\cdots,M-1$. Substituting $\{\mathbf{Q}_j\}$ to Eq. (17), we then obtain the following Eqs. (19) or (20) by which the unknown control points $\{\mathbf{A_I}\}$ are determined.

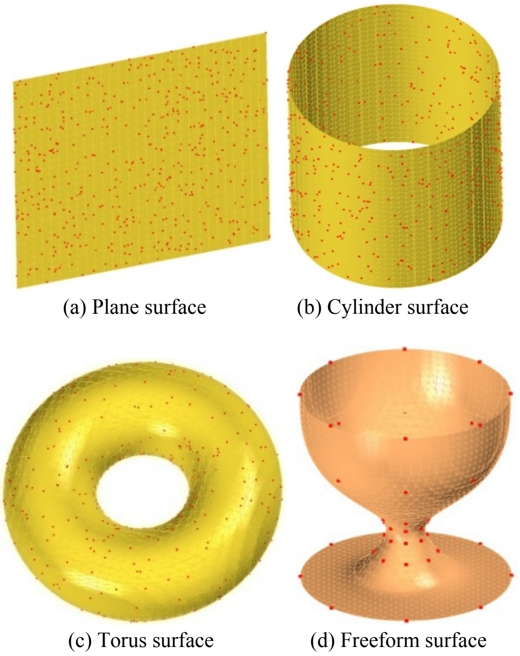
$$\left[\mathbf{N}_{\mathbf{I}}^{i}\right]^{T}\{\mathbf{Q}_{i}\}=\{\mathbf{A}_{\mathbf{I}}\} \tag{19}$$

or

$$\left\{\sum_{i=0}^{M-1}\mathbf{N}_{\mathbf{I}}^{i}\mathbf{Q}_{i}\right\}=\{\mathbf{A}_{\mathbf{I}}\},\quad \mathbf{I}=0,\cdots,L-1 \tag{20}$$

Based on the approximation algorithm above, a B-spline hypervolume can be generated from scattered volume data. We can then approximately extract volume information at an arbitrary position within the scattered volume data for further analysis. We often refer to this generation technique as scattered data modeling [26].

Fig. 1 shows that the proposed B-spline hypervolume enables one to accomplish 3D surface fitting by the approximation algorithm explained above. Fig. 1-(a) to (c) present the construction examples of analytic surfaces (i.e., plane, cylinder, and torus, respectively) as the over-determined constraint cases, where $M = 500$ is used for each example, and $n_1 = n_2 = n_3 = 2$, $k_1 = k_2 = k_3 = 2$ is used for plane surface, $n_1 = n_2 = 4$, $n_3 = 2$, $k_1 = k_2 = 4$, $k_3 = 2$ for cylinder surface, and $n_1 = n_2 = n_3 = 5$, $k_1 = k_2 = k_3 = 5$ for torus surface. Fig. 1-(d) shows the freeform surface example as the under-determined case where $M = 56$, $n_1 = n_2 = n_3 = 4$, and $k_1 = k_2 = k_3 = 4$ are used. In the next section, we describe an interpolation algorithm that generates a B-spline hypervolume from grid-structured volume data.


(a) Plane surface (b) Cylinder surface

(c) Torus surface (d) Freeform surface

Fig. 1. 3D surface fitting examples constructed by the proposed B-spline hypervolume.

4.2 Interpolation of structured data

Once a data set $\left\{\mathbf{P}_{i_1\cdots i_D}\right\}$ (an array of $m_1\times m_2\times\cdots m_D$ points) of a D-dimensional structured grid is given, a rational B-spline hypervolume is constructed by the following interpolation algorithm. Assuming that all homogeneous coordinates equal 1.0, we can develop the following equations for deriving the interpolation algorithm. That is,

$$\begin{aligned}&\mathbf{A}(u_1,\cdots,u_D)\\&=\sum_{i_1=0}^{n_1-1}\cdots\left(\sum_{i_{D-1}=0}^{n_{D-1}-1}\left(\sum_{i_D=0}^{n_D-1}\mathbf{A}^{0}_{i_1\cdots i_D}N_{i_D}^{k_D}(u_D)\right)N_{i_{D-1}}^{k_{D-1}}(u_{D-1})\right)\cdots N_{i_1}^{k_1}(u_1)\\&=\sum_{i_1=0}^{n_1-1}\cdots\left(\sum_{i_{D-1}=0}^{n_{D-1}-1}\mathbf{A}^{1}_{i_1\cdots i_{D-1}}(u_D)N_{i_{D-1}}^{k_{D-1}}(u_{D-1})\right)\cdots N_{i_1}^{k_1}(u_1)\\&\cdots\cdots\cdots\\&=\sum_{i_1=0}^{n_1-1}\mathbf{A}^{D-1}_{i_1}(u_2,\cdots,u_D)N_{i_1}^{k_1}(u_1)\end{aligned} \tag{21}$$

where

$$\mathbf{A}^{0}_{i_1\cdots i_D}=\mathbf{A}_{i_1\cdots i_D} \tag{22}$$

$$\mathbf{A}^{1}_{i_1\cdots i_{D-1}}(u_D)=\sum_{i_D=0}^{n_D-1}\mathbf{A}^{0}_{i_1\cdots i_D}N_{i_D}^{k_D}(u_D) \tag{23}$$

$$\cdots\cdots\cdots$$

$$\mathbf{A}^{D-1}_{i_1}(u_2,\cdots,u_D)=\sum_{i_2=0}^{n_2-1}\mathbf{A}^{D-2}_{i_1 i_2}(u_3,\cdots,u_D)N_{i_2}^{k_2}(u_2) \tag{24}$$

We first determine the D knot vectors of $\mathbf{A}(u_1,\cdots,u_D)$ from the structured volume data. Here, the knot vectors are all nonperiodic (and nonuniform), which are expressed as the following one-dimensional arrays, $\mathbf{U}_1=\left\{t_{i_1}^{(1)}\right\}_{i_1=0}^{i_1=n_1+k_1-1}$, … , and $\mathbf{U}_D=\left\{t_{i_D}^{(D)}\right\}_{i_D=0}^{i_D=n_D+k_D-1}$. Let us obtain the first knot vector $\mathbf{U}_1$ by the following way. Consider the i_1-directional data array $\left\{\mathbf{P}_{i_1 i_2\cdots i_D}\right\}_{i_1=0}^{i_1=n_1-1}$ for each of the cases { (i_2, $i_3,\cdots$, i_D)} where [$i_2 = 0,\cdots, n_2$-1], [$i_3 = 0,\cdots, n_3$-1], $\cdots$, and [$i_D = 0,\cdots, n_D$-1]. That is, we can imagine the $n_2\times n_3\times\cdots\times n_D$ virtual B-spline curves along the i_1 direction. For each i_1-directional data array (or each of $n_2\times n_3\times\cdots\times n_D$ virtual curves), we compute $\left\{c_{i_1 i_2\cdots i_D}\right\}_{i_1=0}^{i_1=n_1+k_1-1}$ (or the knot vector of each virtual curve) by using the parameterization technique introduced by Hartley-Judd [27]. Then we obtain $\mathbf{U}_1 = \left\{t_{i_1}^{(1)}\right\}_{i_1=0}^{i_1=n_1+k_1-1}$ by averaging them as follows:

$$t_{i_1}^{(1)} = \frac{\sum_{i_2=0}^{n_2-1} \cdots \sum_{i_D=0}^{n_D-1} c_{i_1 i_2 \cdots i_D}}{n_2 \times n_3 \times \cdots \times n_D} \quad \text{for } i_1 = 0, \cdots, n_1{+}k_1\text{-}1 \tag{25}$$

By the same way, we can obtain the other knot vectors $\mathbf{U}_2, \cdots, \mathbf{U}_D$ along each direction. Next, from the knot vectors computed above, we determine the Greville abscissa, $\left\{(x_{i_1}, x_{i_2}, \cdots, x_{i_D})\right\}$, corresponding to the data points $\left\{\mathbf{P}_{i_1 \cdots i_D}\right\}$; they are computed by

$$\begin{aligned}
x_{i_1} &= \frac{t_{i_1+1}^{(1)} + t_{i_1+2}^{(1)} + \cdots + t_{i_1+k_1-1}^{(1)}}{k_1 - 1}, \quad i_1 = 0, \cdots, n_1 - 1 \\
x_{i_2} &= \frac{t_{i_2+1}^{(2)} + t_{i_2+2}^{(2)} + \cdots + t_{i_2+k_2-1}^{(2)}}{k_2 - 1}, \quad i_2 = 0, \cdots, n_2 - 1 \\
&\cdots\cdots\cdots \\
x_{i_D} &= \frac{t_{i_D+1}^{(D)} + t_{i_D+2}^{(D)} + \cdots + t_{i_D+k_D-1}^{(D)}}{k_D - 1}, \quad i_D = 0, \cdots, n_D - 1
\end{aligned} \tag{26}$$

Now let us calculate the control points $\mathbf{A}_{i_1 \cdots i_D}$ from the Greville abscissa $\left\{(x_{i_1}, x_{i_2}, \cdots, x_{i_D})\right\}$ computed in Eq. (26) and the corresponding data points $\left\{\mathbf{P}_{i_1 \cdots i_D}\right\}$. At first, substituting the Greville abscissa and their data points into Eq. (21), we get

$$\begin{aligned}
&\mathbf{A}(x_{i_1}, \cdots, x_{i_D}) \\
&= \sum_{i_1=0}^{n_1-1} \mathbf{A}_{i_1}^{D-1}(x_{i_2}, \cdots, x_{i_D}) N_{i_1}^{k_1}(x_{i_1}) = \mathbf{P}_{i_1 \cdots i_D}
\end{aligned} \tag{27}$$

Computing the unknown $\mathbf{A}_{i_1}^{D-1}(x_{i_2}, \cdots, x_{i_D})$ from Eq. (27) and then substituting them into Eq. (24), we obtain

$$\begin{aligned}
&\mathbf{A}_{i_1}^{D-1}(x_{i_2}, \cdots, x_{i_D}) \\
&= \sum_{i_2=0}^{n_2-1} \mathbf{A}_{i_1 i_2}^{D-2}(x_{i_3}, \cdots, x_{i_D}) N_{i_2}^{k_2}(x_{i_2})
\end{aligned} \tag{28}$$

In the same way, we determine the unknown $\mathbf{A}_{i_1 i_2}^{D-2}(x_{i_3}, \cdots, x_{i_D})$ from Eq. (28). Repeating this for Eq. (23) and (22) step by step, we finally get the last control points $\mathbf{A}_{i_1 \cdots i_D}$.

5. Applications of a rational B-spline hypervolume

5.1 Numerical grid generation

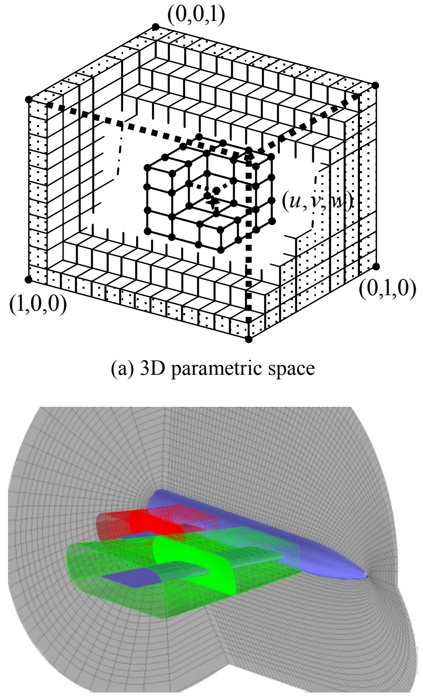
The FDM method is well known as a numerical analysis technique most widely used in CFD applications. The bottleneck in the FDM is the grid generation that produces the structured mesh or curvilinear point set. There have been many researches for improving robustness and performance during grid generation. The Poisson-type elliptic grid generation method [28] is a well-known numerical technique in which the Poisson-type partially-differential equations are repeatedly calculated by using iterative refinement techniques. This method is referred to as numerical grid generation technique [29] since it constructs a grid mesh by moving the finite grid points to satisfy the partially-differential equations. In other words, this is a kind of grid mesh smoothing technique in which each point of the initial grid mesh moves toward desirable directions according to the property of the elliptic partially-differential equations. Note that an initial grid mesh should be given for the numerical generation of the final grid mesh.

Fig. 2 shows initial grid meshes around an airplane, which are generated based on the hypervolume model shown in Eq. (29). In fact, three instances (i.e., hypervolumes) of Eq. (29) are first constructed by the interpolation algorithm in section 4.2 with input points, sampled from 3D Euclidean space around the airplane. A set of parametric points (u, v, w) shown in Fig. 2-(a) are then uniformly sampled from the parametric domain of Eq. (29). Next, the grid points (i.e., nodal points) of each grid mesh are obtained in 3D Euclidean space by simply evaluating Eq. (29) at all parametric points.

One grid mesh is generated from the instance around the airplane body, the others around its two wings. Only two sections of each grid mesh are shown in Fig. 2-(b), which are iso-parametric surfaces calculated from Eq. (29).

$$\begin{aligned}
&\mathbf{A}^{grid}(\mathbf{u}) \\
&= \frac{\sum_i \sum_j \sum_k (hx, hy, hz)_{ijk}^T N_i(u) N_j(v) N_k(w)}{\sum_i \sum_j \sum_k h_{ijk} N_i(u) N_j(v) N_k(w)}
\end{aligned} \tag{29}$$

Note that Eq. (29) maps a bounded parametric cube (represented by u, v, and w in $\mathbf{E}^3$ ($D = 3$)) onto a curvilinear space where point particles are distributed and measured by three geometric attributes, x, y, and z ($K = 3$).



(a) 3D parametric space

(b) 3D Euclidean space

Fig. 2. Grid-structured meshes.

5.2 Flow visualization

Computational flow visualization has been carried out by depicting flow vector field with arrow plots, streamlines, or particle traces. In 2D cases, these graphical methods can be effective to some extent; in 3D cases, however, such visualization can be illegible so that we are likely to overlook flow features or characteristics such as a vortex flow. This is because such techniques only focus on the graphical visualization of flow data without extracting important characteristics implied in the flow data.

To comprehend the flow phenomenon in detail, we thus focus on developing a procedure called feature segmentation algorithm and its graphical techniques [30], which can draw out a characteristic phenomenon from the huge amount of flow data. Developing such algorithms or techniques for analyzing and visualizing fluid flow data requires constructing the flow representation models as like [24].

A rational B-spline hypervolume allows us to build such flow representation models. It can provide computable analysis tools for feature segmentation algorithms while supporting the former visualization techniques. The flow representation model based on the B-spline hypervolume can be represented by Eqs. (29) and (30).

$$\mathbf{A}^{flow}(\mathbf{u}) = \sum_{\mathbf{I}} \left(\rho, \mathbf{V}, T, p, \cdots\right)_{\mathbf{I}}^{T} \mathbf{R}_{\mathbf{I}}(\mathbf{u}) \tag{30}$$

where $\mathbf{A}^{flow}(\mathbf{u})$ describes the distribution of flow field variables (or attributes) in the geometric region described by Eq. (29). The ρ, $\mathbf{V}$, T, and p in Eq. (30) indicate the density, velocity vector (three attributes), temperature, pressure of the fluid flow, respectively. Here, components of a control point in Eq. (30) can be replaced with other flow variables. Note that the two models (Eqs. (29) and (30)) can be merged into one model with nine attributes (K = 9 = 3+6) by taking $(x, y, z, \rho, \mathbf{V}, T, p, \ldots)^{\mathrm{T}}$ as the merged control point if the knot vectors of Eq. (29) are exactly the same as those of Eq. (30). However, it is not recommended to merge models if there is a difference between the geometric complexity and the attribute complexity (e.g., a complex flow in a simple domain).

We now introduce one of the feature parameters used for analyzing a flow structure, and show its computable form based on the proposed hypervolume models, Eqs. (29) and (30). The gradient of a flow velocity $\mathbf{V}$ around the critical point in a flow field, i.e., $\nabla\mathbf{V}$ is a key parameter used frequently for the extraction of flow features or characteristics. By using its eigenvalues and eigenvectors [31, 32], we often investigate a topological structure of a flow field at a certain position. $\nabla\mathbf{V}$ can be calculated from the following expansion shown in Eq. (31).

$$\begin{aligned}\nabla\mathbf{V} &= \left(\frac{\partial\mathbf{V}}{\partial x_1}, \frac{\partial\mathbf{V}}{\partial x_2}, \frac{\partial\mathbf{V}}{\partial x_3}\right)\\ &= \begin{bmatrix} \frac{\partial V_1}{\partial x_1} & \frac{\partial V_1}{\partial x_2} & \frac{\partial V_1}{\partial x_3} \\ \frac{\partial V_2}{\partial x_1} & \frac{\partial V_2}{\partial x_2} & \frac{\partial V_2}{\partial x_3} \\ \frac{\partial V_3}{\partial x_1} & \frac{\partial V_3}{\partial x_2} & \frac{\partial V_3}{\partial x_3} \end{bmatrix} = \left[\frac{\partial V_i}{\partial x_j}\right]\end{aligned} \tag{31}$$

where

$$\left(x_1, x_2, x_3\right) = \left(x, y, z\right), \quad \left(V_1, V_2, V_3\right) = \left(V_x, V_y, V_z\right)$$

$$\frac{\partial V_i}{\partial x_j} = \frac{\partial V_i}{\partial u_1}\frac{\partial u_1}{\partial x_j} + \frac{\partial V_i}{\partial u_2}\frac{\partial u_2}{\partial x_j} + \frac{\partial V_i}{\partial u_3}\frac{\partial u_3}{\partial x_j} = \sum_{k=1}^{3}\frac{\partial V_i}{\partial u_k}\frac{\partial u_k}{\partial x_j}$$

$$\frac{\partial u_k}{\partial x_j} = \frac{\partial u_k / \partial t}{\partial x_j / \partial t} = \frac{1}{V_j}\left(\frac{\partial u_k}{\partial t}\right)$$

$$\frac{\partial u_k}{\partial t} = \frac{\partial u_k}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial u_k}{\partial x_2}\frac{\partial x_2}{\partial t} + \frac{\partial u_k}{\partial x_3}\frac{\partial x_3}{\partial t} = \nabla u_k \cdot \mathbf{V}$$

From above, we finally obtain the computable equation for $\nabla\mathbf{V}$,

$$\frac{\partial V_i}{\partial x_j} = \sum_{k=1}^{3} \frac{\partial V_i}{\partial u_k}\left(\frac{\nabla u_k \cdot \mathbf{V}}{V_j}\right) \quad (i = 1,2,3,\ j = 1,2,3) \quad (32)$$

Here, two terms, V_j and $\dfrac{\partial V_i}{\partial u_k}$ of Eq. (32), can be easily computed by *deBoor* algorithm since they are the zeroth and the first order derivatives of the hypervolume model proposed in Eq. (30), respectively.

Figs. 3 and 4 show the streamlines and stream-surfaces around a curved fan and a blunt fin, respectively, which are typical examples of feature segmentation for feature-based visualization. In these figures, the starting curve of the stream-surfaces is a line and the line becomes curved along its trajectory. The curved trajectory is extracted from the rational B-spline hypervolumes, Eqs. (29) and (30). Although each control point has more than six attributes (x, y, z, V_x, V_y, V_z), we only show the distribution of velocity vectors (each velocity vector has three attributes, V_x, V_y, V_z) around the fan in Figs. 3 and 4. Different colors are used to represent the magnitude of velocity vectors. In addition, Fig. 4 shows a vortex flow (flow feature) at the contact between the fin and the floor. This vortex feature can be also extracted from the proposed hypervolume models. These examples show that the proposed hypervolume models enable one to compute and visualize flow features, implied in a discrete flow data.

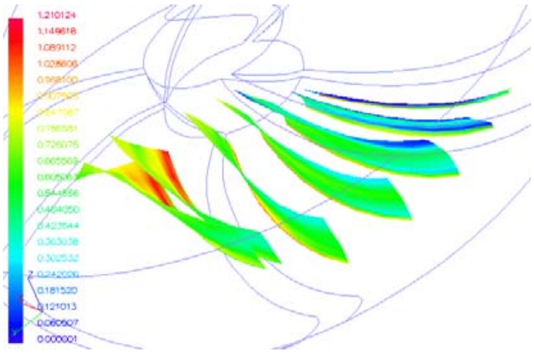


Fig. 3. Stream-surfaces around a curved fin.

5.3 Implicit surface modeling

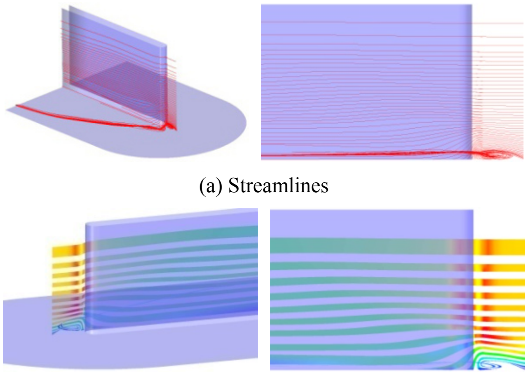
On the basis of the proposed rational B-spline hypervolume, we are able to conduct various implicit surface modelings. Even though a B-spline hypervolume is definitely a parametric representation, it behaves as an implicit surface. This is because we can judge the inside, the outside, and the boundary of an object at an arbitrary position by using the sign of the B-spline hypervolume. That is, an implicit surface can be expressed as the following B-spline hypervolume model.

$$\mathbf{A}^{implicit}(\mathbf{u}) = \sum_{\mathbf{I}} (x, y, z, d)_{\mathbf{I}}^{T} \mathbf{R}_{\mathbf{I}}(\mathbf{u}) \quad (33)$$

where $d_{\mathbf{I}}$ is a distance-like scalar value, typically called an algebraic distance.

A discrete volume data is first constructed by evaluating signed distance values at sample points randomly chosen around the implicit surface. A positive distance value is assigned to the point inside the surface, a negative value to the point outside, and a zero value to the boundary point. A B-spline hypervolume model of Eq. (33) is then generated from the volume data by applying the approximation method proposed in section 4.1. We now can distinguish the sign of the distance value at an arbitrary position and thus determine whether it is inside, outside, or on the target object. Furthermore, Eq. (33) can sufficiently describe level set surfaces [33] which would be a generalized form of an implicit surface, as shown in Fig. 5.

The level set surfaces of a sphere shown in Fig. 5 are generated by the following procedure. At first, we sample the data points which are randomly scattered,


(a) Streamlines

(b) Stream-surfaces

Fig. 4. Stream-surfaces around a blunt fin.

and determine the signed distance value at each point from the distance between the point and the center of the sphere. And then we generate the B-spline hypervolume model of Eq. (33) from the sampling points and their distance values. Finally, we extract the level set surfaces at certain levels from the generated hypervolume model and visualize them by means of marching cubes algorithm [17].

The set-theoretic operations are the primary tasks of the implicit surface modeling. These operations are concerned with the construction of the union, the intersection, or the difference between two implicit surfaces where each operation is principally defined by simple algebraic relations [18]. Modeling tasks described frequently in the field of implicit surfaces can be easily accomplished with the B-spline hypervolumes defined by Eq. (33). Fig. 6 illustrates typical examples of the union, the intersection and the difference between two spheres shown in Fig. 5. The union object shown in Fig. 6-(a) is expressed by a constructive tree like CSG or F-rep in which the terminal nodes contain the implicit spheres shown in Fig. 5 and the root node includes the union operator [18].

From the results so far, we make sure that a rational B-spline hypervolume is a novel representation to participate in implicit surface modeling. This is be cause it behaves as an implicit surface even though it is a parametric representation model.

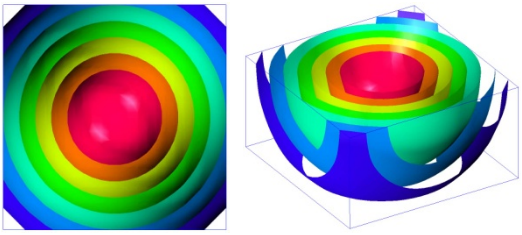
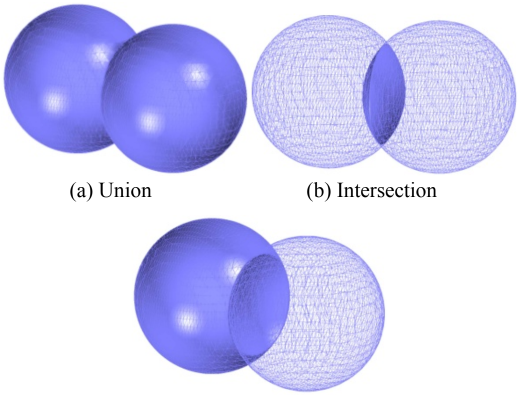


Fig. 5. Level set surfaces of a sphere shape.



(a) Union

(b) Intersection

(c) Difference

Fig. 6. Set-theoretic operations between two spheres.

5.4 Image morphing

Image morphing [34] is a procedure that makes and shows the intermediate images existing between two input images sequentially, by appropriate image modification and mixture. Generally, we call it image metamorphosis, which is well known for the special visualization effects in a movie and TV.

In this paper, we propose an image morphing technique based on the popular mesh warping algorithm [35], in which the image modification and the mixture between two images are represented and manipulated by a rational B-spline hypervolume representation. Following is the procedure of this work. We first choose some distinctive points on two input images, source image and target image, and define the corresponding relation between each pair of them. These relations are then used as input data to generate a space mapping function of Eq. (34) by the interpolation algorithm described in section 4.2.

$$\begin{aligned}\mathbf{A}^{mapping}(\mathbf{u}) &= (x_s, y_s, x_t, y_t) \\ &= \sum_{\mathbf{I}} (x_s, y_s, x_t, y_t)_{\mathbf{I}}^T \mathbf{R}_{\mathbf{I}}(\mathbf{u})\end{aligned} \tag{34}$$

where s and t are subscripts denoting the source and the target images, respectively. (x, y) denotes the image pixel coordinates, and $\mathbf{u} = (u, v)$ denotes the parametric domain of two images. Note that the input data is given in the form of 6-dimensional array,

$$\mathbf{P} = \left\{ (u, v, x_s, y_s, x_t, y_t)_i \;\middle|\; \begin{array}{l} (u, v)_i \in \mathbf{E}^2, \\ (x_s, y_s, x_t, y_t)_i \in \mathbf{E}^4 \end{array} \right\} \tag{35}$$
$$(i = 0, \cdots, M-1)$$

where $(u, v)_i$ is the data position of Eq. (34) and $(x_s, y_s, x_t, y_t)_i$ is its function value. The space mapping function of Eq. (34) is used to define a mutual mapping between the source pixel coordinates and the target ones. The parameters (u, v) are first searched from either of the two pixel coordinates, and then the other pixel coordinates are obtained by evaluating Eq. (34) at the searched parameters. We set $(u, v) = (x_t, y_t)$ for fast computation.

Next, warped images are obtained by the space mapping explained above, and then both the unwarped and warped images are interpolated to make the image warping function shown in Eq. (36).

$$\mathbf{A}^{warping}(\mathbf{u}) = \sum_{\mathbf{I}} (x, y, z, r, g, b, \alpha)_{\mathbf{I}}^{T} \mathbf{R}_{\mathbf{I}}(\mathbf{u}) \tag{36}$$

where r, g, b denote the colors of the images, and α denotes the opacity value. The images before and after modification are located at $z = 0$ and $z = 1$, respectively. The parameters (u, v) in the $\mathbf{u} = (u, v, w)$ of Eq. (36) are used to represent the 2-D image plane, while w is the parameter denoting the modifying direction of the image. For example, when $0 \le w \le 1$ in Eq. (36), $\mathbf{A}^{warping}$ $(u, v, w = 0.5)$ shows the intermediate image between the before image ($w = 0$) and the after image ($w = 1$).

We generate the image warping function from each of two input images, and then produce the final morphing function of Eq. (37) by the weighted sum of the two image warping functions.

$$\begin{aligned} &\mathbf{A}^{morphing}(\mathbf{u}) \\ &= (1-w)\cdot \mathbf{A}_{s}^{warping}(u,v,w) + w \cdot \mathbf{A}_{t}^{warping}(u,v,1-w) \end{aligned} \tag{37}$$

where $\mathbf{A}_{s}^{warping}(\mathbf{u})$ and $\mathbf{A}_{t}^{warping}(\mathbf{u})$ are the image warping functions that define the image warping of the source and the target, respectively. w is a weighting factor for the weighted sum and the direction of morphing process at the same time.

Fig. 7 shows the primitive form of image morphing. Some successive cross-sectional images of the morphing function are shown when the source and the target images are linearly interpolated without the image warping achieved by Eq. (34) and (36). It is clearly shown that the two images are overlapped at the middle of the process while progressing from the top-left to the bottom-right.

Fig. 8 illustrates the whole process of image morphing by Eqs. (34), (36) and (37). The images in the first and the second rows show the warping of the source image that progresses from left to right in Fig. 8, while the images in the fourth and the fifth rows show the warping of the target image that progresses from right to left in Fig. 8. Here, the white grids on the images denote the input data, shown in Eq. (35), of which warping progress is also displayed together. The images in the third row show the weighted average of the warping images in the second and the fourth rows by Eq. (37), which is the final process of the image morphing.

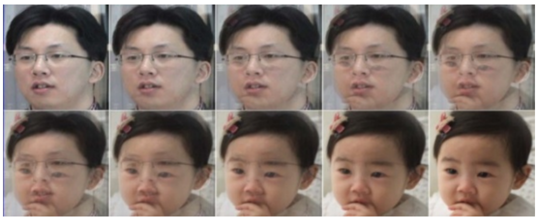


Fig. 7. Image morphing without image warping.

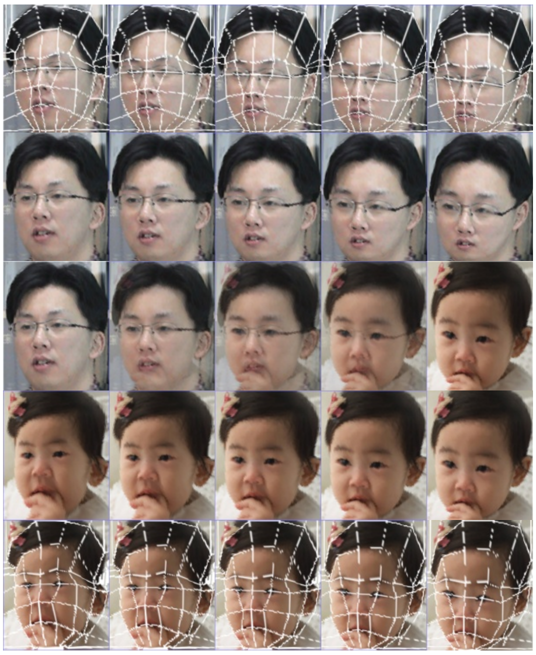


Fig. 8. Image morphing with image warping.

As mentioned above, a rational B-spline hypervolume can produce the morphing images existing in the 3-D parametric space, which consists of 2-D image space and 1-D morphing (or warping) direction. That is, it is said that the former descriptive algorithm of image morphing can be formulated and calculated by a rational B-spline hypervolume in a mathematical way.

6. Conclusions

This paper presents a rational B-spline hypervolume representation that is capable of describing a closed volume object with K attributes embedded in D-dimensional space. We also propose approximation and interpolation algorithms for generating a rational B-spline hypervolume from the given volume data distributed over a multidimensional space. In addition, useful expressions for computing differential elements and derivative operators are derived (not presented in this paper) to figure out the complex structure of general volume objects.

Moreover, we show that a rational B-spline hypervolume can be utilized as modeling and analysis tools applicable to various applications. The example of grid generation shows that the rectilinear parameter space of a B-spline hypervolume is mapped into 3D curvilinear coordinates so that grid meshes can be easily generated. The example of flow visualization proves that a B-spline hypervolume is a good mathematical tool by which the flow features or characteristics can be analyzed and visualized. The example of implicit surface modeling provides the possibility toward a dual representation in both implicit and parametric forms. The example of image morphing shows the capability to represent a warping and morphing process.

After all, we make sure that a rational B-spline hypervolume is a generalized volume representation without any dependence on specific applications and is capable of carrying out the modeling, analysis, and visualization of a multivariate volume object defined in a multidimensional space.

As a future research topic, we need to study hierarchical structures based on a rational B-spline hypervolume to represent and manage the partially local complexity or the multiresolution of a volume object effectively. Now, we are examining the wavelet-type multiresolution model by using a knot insertion algorithm based on a rational B-spline hypervolume. In addition, we are carefully inspecting that the proposed hypervolume can be used as a solution structure for meshless methods [36, 37] and meshfree modeling and simulations [38]. That is, we are investigating that a rational B-spline hypervolume could offer an approximate solution to governing differential equations (or equivalent variational form) as an admissible trial function for boundary-value problems. Substituting a rational B-spline hypervolume into the differential equation where the B-spline hypervolume satisfies the boundary conditions, we could derive the error residuals formulated with the unknown control points of the rational B-spline hypervolume. Here, the geometric part of the control point would play the role of a nodal point of a mesh element [39], and the field variables at each node could be represented by the non-geometric part of the control point.

Acknowledgment

The research was supported by a grant from the University Restructuring Program (funded by the Ministry of Education and Human Resources Development) of Chungju National University, and by the Korea Research Foundation Grant funded by Korean Government (MOEHRD) (KRF-2004-202-D00075).

References

[1] T. T. Elvins, A survey of algorithms for volume visualization, *Computer Graphics*, 26 (3) (1992) 194-201.

[2] A. E. Kaufman, State-of-the-art in volume graphics, In: M. Chen, A. E. Kaufman, and R. Yagel, editors, *Volume Graphics*, Springer, (2000).

[3] G. M. Nielson, Volume modeling, In: M. Chen, A. E. Kaufman, and R. Yagel, editors, *Volume Graphics*, Springer, (2000).

[4] M. Chen, A. S. Winter, D. Rodgman and S. M. F. Treavett, Enriching volume modelling with scalar fields, In: F. Post, G-P Bonneau, and G. Nielson, editors, *Data Visualization: The State of The Art*, Kluwer Academic Press, (2002).

[5] L. Hesselink, Research issues in vector and tensor field visualization, *IEEE Computer Graphics and Applications*, 14 (2) (1994) 50-56.

[6] R. Franke and G. M. Nielson, Smooth interpolation of large sets of scattered data, *International Journal for Numerical Methods in Engineering*, 15 (1980) 1691-1704.

[7] R. Franke and G. M. Nielson, Scattered data interpolation and applications: A tutorial and survey, In: H. Hagen, and D. Roller, editors, *Geometric modeling: methods and their application*, Springer-Verlag, Berlin, (1990) 131-160.

[8] G. M. Nielson, et al, Visualization and modeling of scattered multivariate data, *IEEE Computer Graphics and Applications*, 11 (3) (1991) 47-55.

[9] G. M. Nielson, A method for interpolating scattered data based upon a minimum norm network, *Mathematics of Computation*, 40 (1983) 253-271.

[10] C. Bajaj, V. Pascucci, G. Rabbiolo, and D. Schikore, Hypervolume visualization: a challenge in simplicity, *IEEE Symposium on Volume Visualization*, ACM SIGGRAPH (1998) 95-102.

[11] H. Bruce, McCormick, A. Thomas, DeFanti, and D. B. Maxine, Visualization in scientific computing, *Computer Graphics*, 21 (6) (1987) 1-14.

[12] L. Piegl and W. Tiller, *The NURBS Book*, Springer-Verlag, (1995).

[13] J. Wilhelms and J. Challinger, Direct volume rendering of curvilinear volumes, *Computer Graphics* (*San Diego Workshop on Volume Visualization)*, 24 (5) (1990) 41-47.

[14] S. Silva, J. S. B. Mitchell and A. E. Kaufman, Fast rendering of irregular grids, *Volume Visualization Symposium*, October (1996) 15-22.

[15] J. F. Blinn, A generalization of algebraic surface drawing, *ACM Transactions on Graphics*, 1 (3) (1982) 235-256.

[16] J. Bloomenthal, Polygonization of implicit surfaces, *Computer Aided Geometric Design*, 5 (4) (1988) 341-355.

[17] W. E. Lorensen and H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, *Computer Graphics* (*Proceedings of SIGGRAPH 87)*, 21 (4) (1987) 163-169.

[18] A. Pasko, V. Adzhiev, A. Sourin and V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer*, 11 (8) (1995) 429-446.

[19] A. A. G. Requicha, Representations for rigid solids: theory, methods and systems, *ACM Computing Surveys*, 12 (4) (1980) 437-464.

[20] S. Fang, R. Srinivasan, and S. Venkataraman, Volumetric CSG - A model-based volume visualization approach, *Sixth International Conference in Central Europe on Computer Graphics and Visualization*, February (1998).

[21] M. Chen and J. V. Tucker, Constructive volume geometry, *Computer Graphics Forum*, 19 (4) (2000) 281-293.

[22] A. Pasko, V. Adzhiev and B. Schmitt, Constructive hypervolume modeling, *Technical Report TR-NCCA-2001-01*, National Centre for Computer Animation, Bournemouth University, UK, February (2001).

[23] C. de Boor, *A practical guide to splines*, Springer - Verlag, (1978).

[24] S. Park and K. Lee, High-dimensional trivariate NURBS representation for analyzing and visualizing fluid flow data, *Computers & Graphics*, 21 (4) (1997) 473-482.

[25] C. de Boor, On calculating with B-splines, *Journal of Approximation Theory*, 6 (1972) 50-62.

[26] G. M. Nielson, Scattered Data Modeling, *IEEE Computer Graphics and Applications*, 12 (1) (1993) 60-70.

[27] P. J. Hartley and C. J. Judd, Parameterization and shape of B-spline curves for CAD, *Computer-Aided Design*, 12 (5) (1980) 235-238.

[28] W. Schwartz, Elliptic grid generation system for three-dimensional configuration using Poisson's equation. In: J. Hauser and C. Taylor, editors, *Numerical grid generation in computational fluid dynamics*, Pineridge Press Limited, (1986).

[29] J. F. Thompson, Z.U.A. Warsi and C. Mastin, *Numerical grid generation foundation and applications*, Elsevier Science Publishing, (1985).

[30] A. Sanna, B. Montrucchio and P. Montuschi, A survey on visualization of vector fields by texture-based methods, *Research Developments in Pattern Recognition*, 1 (1) (2000).

[31] A. Globus, C. Levit, and T. Lasinski, A tool for visualizing the topology of three-dimensional vector fields, *Proceedings IEEE Visualization*, (1991) 33-40.

[32] J. Helman and L. Hesselink, Visualizing vector field topology in fluid flows, *IEEE Computer Graphics & Applications*, 11 (3) (1991) 36-46.

[33] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, UK, (1999).

[34] S. Lee, K-Y Chwa, J. Hahn and S. Y. Shin, Image morphing using deformation techniques, *J. Visualization and Computer Animation*, 7 (1) (1996) 3-23.

[35] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos, CA, (1990).

[36] X. Zhang, M. Rayasam and G. Subbarayan, A meshless, compositional approach to shape optimal design, *Computer methods in applied mechanics and engineering*, 196 (2007) 2130-2146.

[37] T. J. R. Hughes, J. A. Cottrell and Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer methods in applied mechanics and engineering*, 194 (2005) 4135-4195.

[38] I. Tsukanov and V. Shapiro, Meshfree modeling and analysis of physical fields in heterogeneous media, *Journal of Advances in Computational Mathematics*, 23 (1-2) (2005) 95-124.

[39] P. Yang and X. Qian, A B-spline-based approach to heterogeneous objects design and analysis, *Computer-Aided Design*, 39 (2007) 95-111.

Sangkun Park received his B.S. in Mechanical Engineering from POSTECH, Korea, in 1991. He then received his M.S. and Ph.D. degrees from Seoul National University in 1993 and 1997, respectively. Dr. Park is currently a Professor in Mechanical En-gineering at Chungju National University in Chungju, Korea. His research interests include geometric modeling and processing, CAD/CAM/ CAE, scientific visualization, virtual engineering, and structural optimization.