Solid Fitting: Field Interval Analysis for Effective Volume Exploration

Issei Fujishiro and Yuriko Takeshima[†]

Department of Information Sciences, Ochanomizu University
† Doctoral Research Course in Human Culture, Ochanomizu University

(Received December 11, 1998)

Abstract

In previous reports, the concept of solid fitting has been presented as a new indirect approach to volume visualization. Solid fitting relies on a simple, but powerful geometric data model, termed interval volume, that allows one to represent a three-dimensional subvolume for which the associated scalar values lie within a specified closed interval. This paper combines the latest results obtained through the course of the solid fitting project. After reviewing the salient features of interval volume and the fundamentals of solid fitting in the first two sections, Section 3 discusses improvements to the original solid fitting algorithm so as to extract interval volumes in a topologically-consistent manner. Also, the octree-based acceleration mechanism incorporated into the algorithm is analyzed further with a complex, time-evolving, volumetric data set. Section 4 is devoted to the presentation of several representative operations related to interval volume, including focusing and measurement-coupled visualization. In addition, a candidate for the volumetric coherence measure is introduced for adaptive solid fitting and its application to multi-scalar visualization. Lastly, the paper is summarized with some remarks on a hybrid volume exploration environment, in which solid fitting plays various roles.

1 Introduction

In previous reports[4, 5], the concept of *solid fitting* has been proposed to present a new *indirect* approach to *volume visualization*[11]. Solid fitting relies on a simple, but powerful geometric data model, termed *interval volume*, that allows one to represent a three-dimensional(3D) subvolume for which the associated scalar values lie within a specified closed interval. Geometric modeling of field interval-based volumetric regions of interest(ROIs) has recently begun to attract much attention from visualization researchers. For instance, Udupa proposes a cuberille model, called *shell*, for the compact representation and fast manipulation of regular volumetric data sets containing fuzzy boundaries between adjacent material[23]. Guo presents *interval sets* with the aim of unifying *surface fitting*(SF) and *direct volume rendering*(DVR) approaches[9]. Crawfis exploits the concept of *data space slicing* in the context of realtime interaction with volumetric objects in a virtual environment[3].

Interval volume can be viewed as a more effective tool for exploring volumetric ROIs, compared with the traditional SF approach, which uses *isosurfaces* with specified field values. Consider, for example, a 64 × 64 × 64 theoretical volume data set for the 3D electron density distribution around an H_2 molecule[26]. As clearly seen from Fig.1, introducing the interval-based specification of ROIs leads to not only the producibility of more intuitive and informative images through boundary surface transparency and orthogonal slicing, but also the computability of quantities such as the surface area, total volume, and field average over the ROIs. DVR algorithms can generate a semi-transparent image of the entire volumetric data set for one to peer inside to, without the aid of intermediate geometrical representations. However, DVR algorithms are inherently computationally-expensive, in spite of many efforts to reduce calculation time by algorithmic optimization, usage of parallel computation, and development of special hardware architectures. On the other hand, for the case of a spatially-coherent data set just as in Fig.1, the solid representation of volumetric ROIs is more spatially-efficient than the original lattice structure, hence promising economical data storage and transmission. As in the SF approach, well-tuned surface rendering software and graphics engines can also be utilized for interactive display and manipulation of interval volumes.

Furthermore, in practical situations, structural ambiguity inherent to natural objects, or error due to measurement, mathematical modeling or numerical computation is frequently expected to exist in a target volumetric data set. If these error sources are not fully taken into account, extracted isosurfaces may convey little meaningful information for exploratory purposes. Thus, geometric structures to be

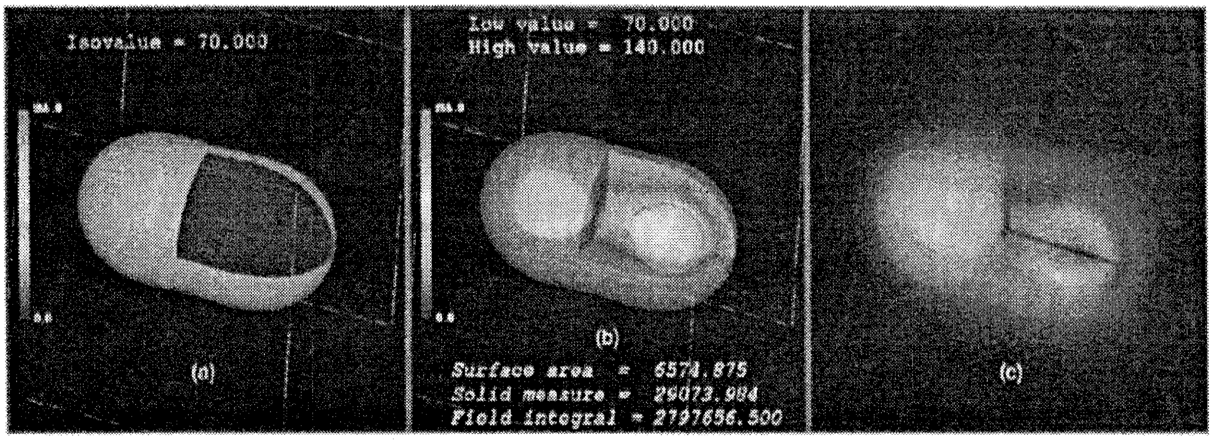

Figure 1: Three volume visualization approaches for visualizing the 3D electron density distribution around an H_2 molecule. (**a**) Surface fitting; (**b**) solid fitting; and (**c**) direct volume rendering.

extracted from practical volumetric data sets are required to tolerate some variation in field values.

For instance, consider an analytical volumetric function defined by:

$$f(x, y, z) = \frac{(2^8-1)}{92}\{x^3 + y^3 + \frac{3}{2}z^2(x + y + 1) - 3xy\} + \frac{2^8-1}{2} \qquad (|x, y, z| \leq 2).$$

Indeed, this function is a 3D version of *Folium of Descartes*, whose implicit surface defined by the equation $f(x, y, z) = 127.5 = (2^8-1)/2$ has the origin of the volume as its singular point. Fig.2(a) shows five isosurfaces extracted from the volume. Notice that due to digitization error, the third isosurface whose field value is 127.5 splits into two parts, and does not pass the center(origin) of the volume. This may give a misleading interpretation about the topological feature of the volumetric function. On the other hand, Fig.2(b) shows that interval volume, having a very thin interval around 127.5, retains the correct topology of the target isosurface, from which the existence of the singular point can be easily imagined. This suggests another attractive use of interval volume as a *generalized* isosurface.

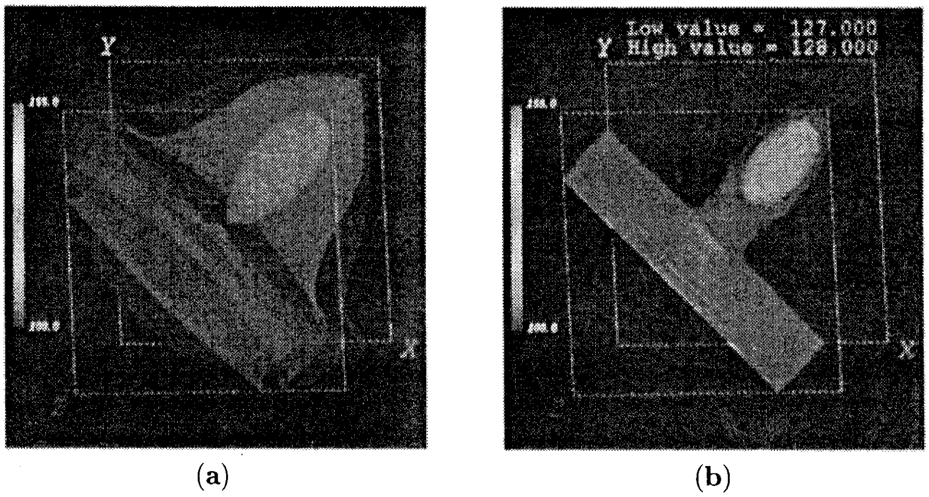

Figure 2: Visualization of 3D Folium of Descartes. (**a**) Surface fitting (five isosurfaces whose target fields are 102.5, 115, 127.5, 140, and 152.5); and (**b**) solid fitting (interval volume whose field interval is [127.0, 128.0]).

This paper combines the latest results obtained through the course of the solid fitting project[6, 7, 20, 22]. After reviewing the fundamentals of solid fitting in the next section, Sect.3 discusses improvements to the original solid fitting algorithm so as to extract a high-resolution, polyhedral solid data structure of interval volume in a topologically-consistent manner[20]. Also, the octree-based acceleration mechanism[5] incorporated into the algorithm is analyzed further with a complex, time-evolving, volumetric data set[6]. Sect.4 is devoted to the presentation of several representative operations related to interval volume, including focusing and measurement-coupled visualization[5]. In addition, a candidate for the volumetric coherence measure is introduced for adaptive interval volume extraction and its application to multi-scalar visualization. Lastly, the paper is summarized with some remarks on a hybrid volume exploration environment, in which solid fitting plays various roles.

2 Fundamentals of Interval Volume

2.1 Interval volume definition

Consider that a continuous source volume space $V \equiv (S, f)$ is discretized and quantized into a *regular* voxel data set V_D[0], where S denotes a rectangular 3D region in R^3, and f denotes a scalar *field function(field* hereafter), which maps S into a finite closed interval $[f_{min}, f_{max}](\subset R)$. In the remainder of this paper, the term *voxel* is defined as a sample point which is located at a grid coordinate, and has a single value for associated field f. Eight adjacent voxels constitute an axis-aligned rectangular prism, called a *cube*(Fig.3). Note that the total number of cubes, N, is given by:

$$N = n_x \times n_y \times n_z. \tag{1}$$

Unless explicitly noted, the trilinear interpolation scheme is used to reconstruct a field value $f(p)$ at an arbitrary point $p \in S$ from the field values at the vertex voxels of the cube to which p belongs.

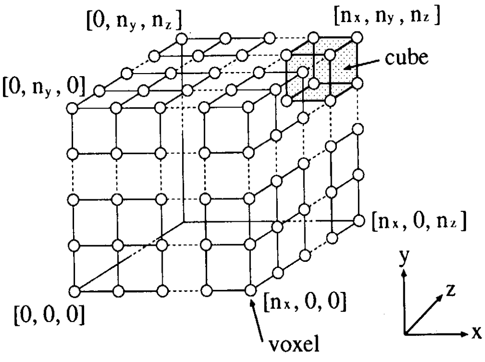

Figure 3: Definitions in source volume V_D.

Interval volume $IV(\alpha, \beta)$ is defined as a subset of the given volume space V such that for any $p \in S$, the field value $f(p)$ belongs to a single finite closed interval $[\alpha, \beta]$. Specifically,

$$IV(\alpha, \beta) = \{(p, f(p)) | p \in S, \alpha \leq f(p) \leq \beta\} \qquad (f_{min} \leq \alpha,\ \beta \leq f_{max}).$$

The main aim here is to extract from the source volume data set V_D, a high-resolution, discrete interval volume IV_D. Here, the word "high-resolution" means that IV_D retains certain geometric structures at the subcube level[12]. Under the trilinear field assumption, IV_D can become a polyhedral *solid*, which is represented by a set of directed polygonal patches, retaining the original field values at voxels located inside the IV_D. The polygonal information can be managed systematically using a simple vertex-based boundary representation[13].

Since interval volume is a subvolume, it does not suffer from the SF constraint that ROIs are limited to lie on 3D surfaces. Therefore, interval volume $IV(\alpha, \beta)$ has different interpretations according to how its interval $[\alpha, \beta]$ is specified. For more detailed discussions, see [4, 5], which describe a useful application of interval volume having a thin interval to visualization of a 3D Gaussian field containing a virtual noise assumed to arise during the phase of volumetric data acquisition.

[0] The results shown below can be extended to *structured(curvilinear)* data sets in a straight-forward manner. Recently, further extension to the *unstructured* case has been achieved by the invention of an interval volume tetrahedrization algorithm[15].

2.2 Solid fitting with interval volume

In order to extract an interval volume $IV_D(\alpha, \beta)$ from a source volume V_D, the *Marching Cubes*(MC) isosurfacing algorithm[12] is extended[4, 5]. The extended MC algorithm is designed so as to determine a proper *polyhedral block pattern* for each cube containing a target interval volume. The actual approach taken in this paper is to perform a set operation on two special kinds of interval volumes in a cubewise manner.

The algorithm can be summarized with the following two primary steps:

For each cube i,

Step1: *Refer to the modified look-up table to generate polyhedral block regions α-cube $C^i_{low}(\alpha)$ and β-cube $C^i_{high}(\beta)$, representing $IV_D(\alpha, f_{max})$ and $IV_D(f_{min}, \beta)$, respectively.*

Step2: *Evaluate the intersection of these blocks to yield $IV_D(\alpha, \beta)$ in each cube, and take the union of them over the entire volume:*

$$IV_D(\alpha, \beta) = \bigcup_{i=1}^{N} \left(C^i_{low}(\alpha) \cap C^i_{high}(\beta)\right), \tag{2}$$

*where N denotes the total number of cubes*Eq.(1).

The number of polyhedral block pattern entries for both special kinds of cubes is clearly the same as that of polygonal patch pattern entries used in the original MC algorithm. Therefore, the temporal complexity of the MC-based solid fitting algorithm is expected to be on the same order $O(N)$ as that of the original MC algorithm.

For brevity, a two-dimensional(2D) example[1] is first used to describe how to construct polyhedral blocks for interval volume within a single cube. Consider the case that $IV_D(25, 28)$ is to be extracted from the cube shown in Fig.4. Note that the number beside each voxel represents its associated field value. The intermediate-value theorem guarantees that two points with field values 25 and 28 appear on the bottom and right edges. Clearly, α-cube corresponds to the lower-right triangle region, while β-cube to the upper-left trapezoid. The intersection of these regions turns out to be the hatched quadrilateral, which is equivalent to the target $IV_D(25, 28)$ block. The quadrilateral has parts of two cubes' boundaries as its sides, as depicted with solid line segments.

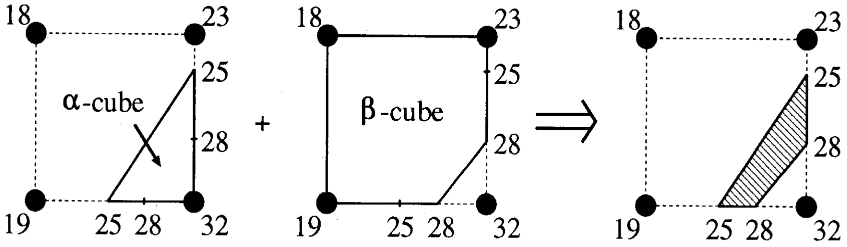

Figure 4: Interval volume extraction in a single cube(2D).

It should be noted here that an analysis of the nature of the trilinear interpolation scheme reveals that, whenever a cube has a face-connected neighbor, all patches on the shared boundary of the cubes can be omitted(Fig.5). If such a boundary patch is not removed, the total number of patches becomes $O(N)$ in the worst case. This offsets the benefit of relatively low spatial/temporal complexities in the indirect volume visualization. The topic of topological consistency of polyhedral blocks for interval volume in adjacent cubes will be discussed in Subsect.3.1.

Lastly, Fig.6 shows the complementary α-cubes and β-cubes for fifteen, common, polyhedral block patterns[5]. These fifteen patterns correspond to those for isosurfaces used in the original MC algorithm[12]. Note that for each pattern, the two complementary cases are required to be distinguished according to whether vertex field values are above (or below) either of the specified interval limit values.

[1] Related 3D terms such as cube and isosurface are used in 2D examples.

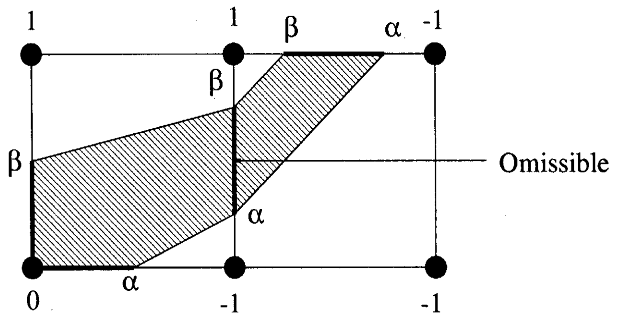

Figure 5: Interval volume extraction in adjacent cubes(2D). The number beside a vertex v shows the result of voxel classification in terms of order relationship between a target interval $[\alpha, \beta]$ and the voxel field value $f(v)$. -1: $f(v) < \alpha$; 0: $\alpha \leq f(v) \leq \beta$; and 1: $\beta < f(v)$.

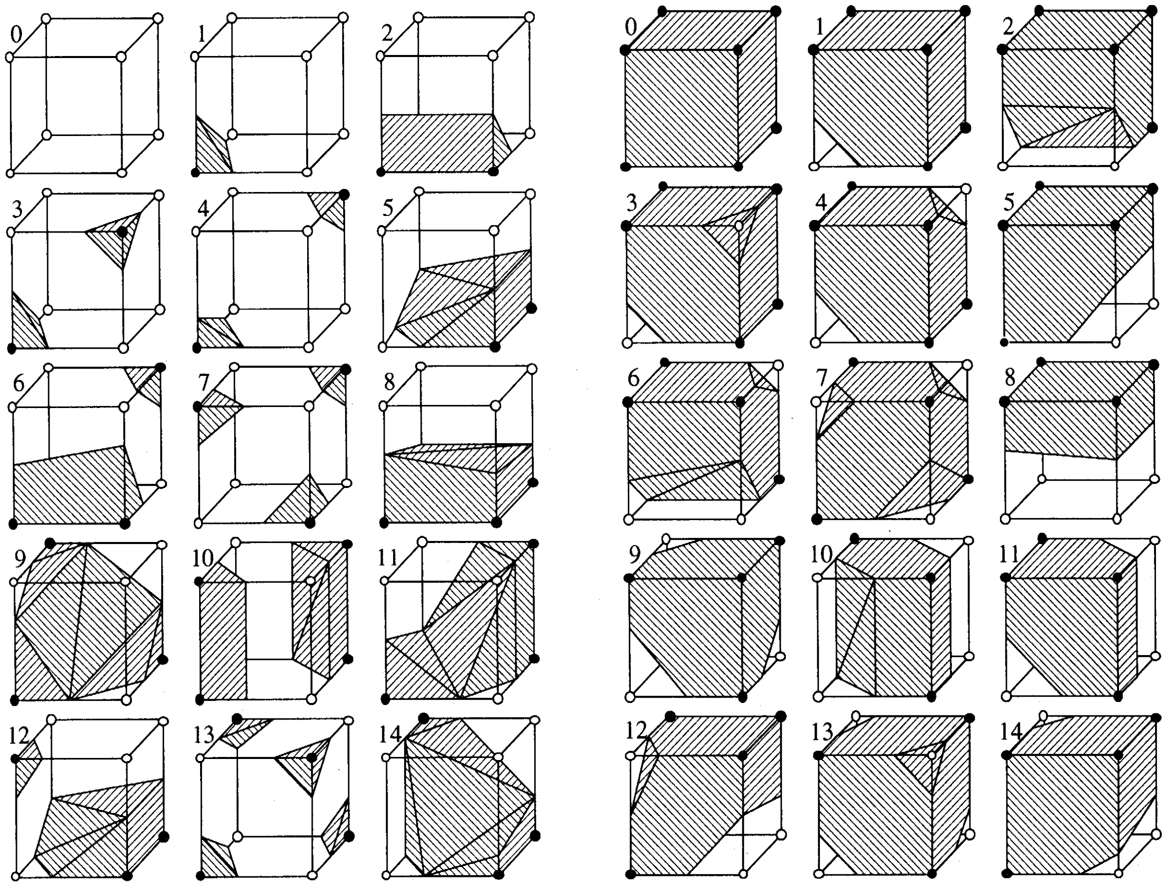

Figure 6: Polyhedral block configuration for α-cube and β-cube (α-cube: $\circ < \alpha \leq \bullet$; β-cube: $\bullet \leq \beta < \circ$).

2.3 Interval volume module

A module called interval volume, executable on a widely used commercial visualization software called AVS5[2][1, 2] has been developed. This module creates an interval volume of a given field interval specification. Fig.7(a) shows a network editor's window, which displays the control panel of the latest version of the module on the left, a typical solid fitting network including the module in the middle workspace, and an output image window of the HIPIP testbed volumetric data set[16] on the right. Fig.7(b) lists hierarchically all control pages and widgets appearing in the module control panel in Fig.7(a). As for the details on the fundamental operability of the control panel, and the dataflow mechanism of the solid fitting network, see [4]. Other new features of the interval volume module will also be described in subsequent subsections.

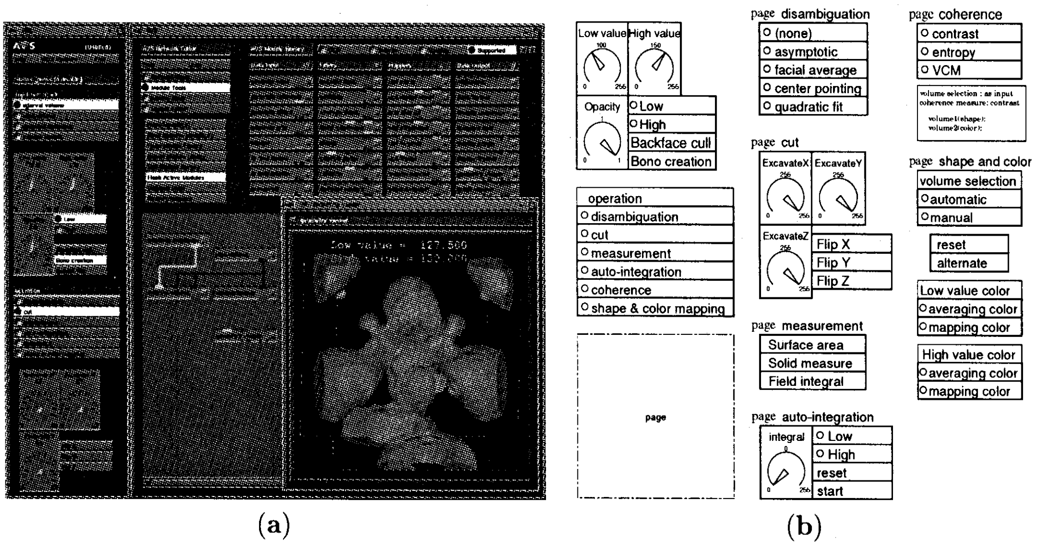

Figure 7: Solid fitting network. (**a**) A typical AVS module network and a resultant HIPIP image; and (**b**) interval volume control pages and widgets.

3 Sophisticated Extraction of Interval Volume

3.1 Disambiguation

3.1.1 Ambiguous cube face

Topological consistency is crucial for geometric structures to be extracted from a given volumetric data set, because incorrect topological features might lead to a misunderstanding of the target data set.

As in the case of isosurfaces, erroneous connection of interval volume polyhedral blocks may occur when the common face of adjacent cubes is *ambiguous.* Ambiguity occurs when the two diagonal pairs of vertex voxels have field values of the same sign with respect to local target interval limit values, and the signs of the vertex pairs are different from each other. Under the trilinear field assumption, there are two possible patterns for the boundary patches of α-cube and β-cube (Fig.8).

A polyhedral block pattern for α-cube and β-cube depicted in Fig.6 is called *ambiguous* if the corresponding cube has one or more ambiguous face. Cases 3, 6, 7, 10, 12, and 13 are such examples. Consider the examples shown in Fig.9, where for both cases 6 and 3, two possible polyhedral block patterns for α-cube are depicted. Note that 6B and 3B are the original patterns, and a rotation is needed to obtain 3B. Clearly, α-cube in 6A(B) and α-cube in 3A(B) can be connected correctly in a topological sense, while connection of α-cube in 6A(B) and α-cube in 3B(A) yields an erroneously connected interval volume.

[2] AVS is a trademark of Advanced Visual Systems, Inc.

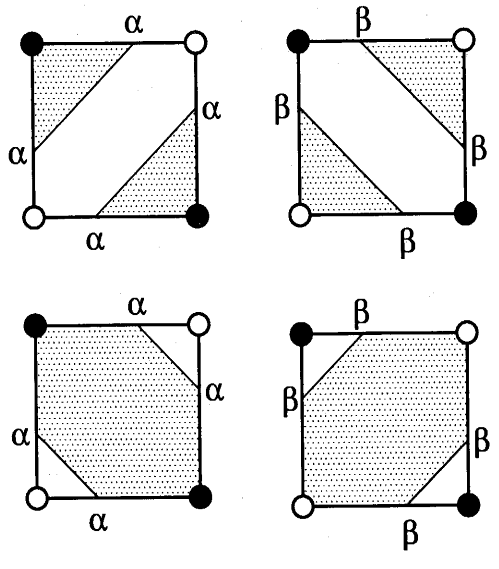

Figure 8: Possible $\alpha(\beta)$-cube patch patterns of interval volume boundary on ambiguous face ($\circ < \alpha \leq \beta \leq \bullet$).

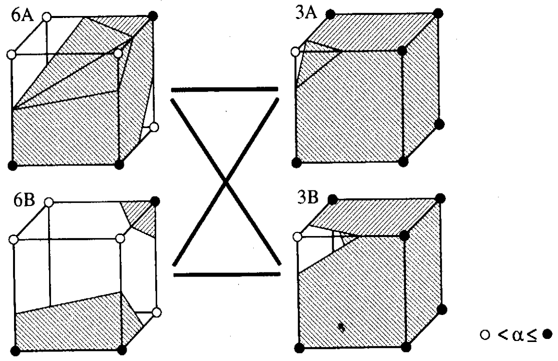

Figure 9: Four possible connections of polyhedral block patterns for ambiguous cases 6(left) and 3(right).

3.1.2 Disambiguation schemes

Herein, the following four known auxiliary algorithms, which alleviate the topological ambiguities of isosurfaces extracted by the MC algorithm, are extended so as to connect polyhedral blocks for interval volume in adjacent cubes in a topologically-consistent manner. The common extension for disambiguating the connection of interval volume polyhedral blocks is to consistently choose either of the two cases for solid region of α-cube and β-cube, based on a specific threshold derived from local field values:

- *Facial Average Values*(FAV)[28]: This algorithm evaluates the center value of a bilinear interpolation across the ambiguous face, and compares it with a target interval to locate the solid region at the proper position.
- *Asymptotic Decider*(AD)[14]: Under the bilinear interpolation assumption, interval volume boundary forms a hyperbolic curve on a cube face. This algorithm improves the quality of the FAV's disambiguation by evaluating the sign at the saddle point of the hyperbolic curve.
- *Gradient Consistency Heuristics*(GCH)[25]: This method utilizes not only vertex field values but also their gradients for more precise field reconstruction based on the *biquadratic* interpolation assumption. Proposed in [25] are two particular algorithms categorized into this method, i.e., *Center Pointing Gradient*(CPG) and *Quadratic Fit*(QF), which are regarded as quadratic extensions to the FAV and AD algorithms, respectively.

One of these disambiguation algorithms should be performed prior to the calculation of cubewise block intersection in Eq.(2). In the control page disambiguation of the interval volume module in Fig.7(b), one of the present disambiguation algorithms can be manually chosen according to the volumetric coherence of a given data set (see Subsect.4.3).

3.1.3 Experiments

First, a 4 × 4 × 4 volumetric data set representing the following two analytical volumetric functions ($0 \leq x, y, z \leq 3$) are used to verify to what degree the four algorithms select proper polyhedral block patterns for interval volume.

$$\begin{aligned} f_1(x,y,z) &= 4(y-1)^2 + 2(x-z)^2 - 2(x+z-3)^2 + 1 \\ f_2(x,y,z) &= 4y + 4(x-z)^2 - 5 \end{aligned}$$

Fig.10 extracts $IV_D(25, 30)$[3] for each function using the present auxiliary algorithms. In both functions, topologically-consistent interval volumes are extracted as control cases using a recursive subdivision algorithm. It is seen from the figures that, as for f_1, all four disambiguation algorithms and even the original solid fitting algorithm without any disambiguation yield the common and correct polyhedral blocks, but that, as for f_2, only the two GCH algorithms do.

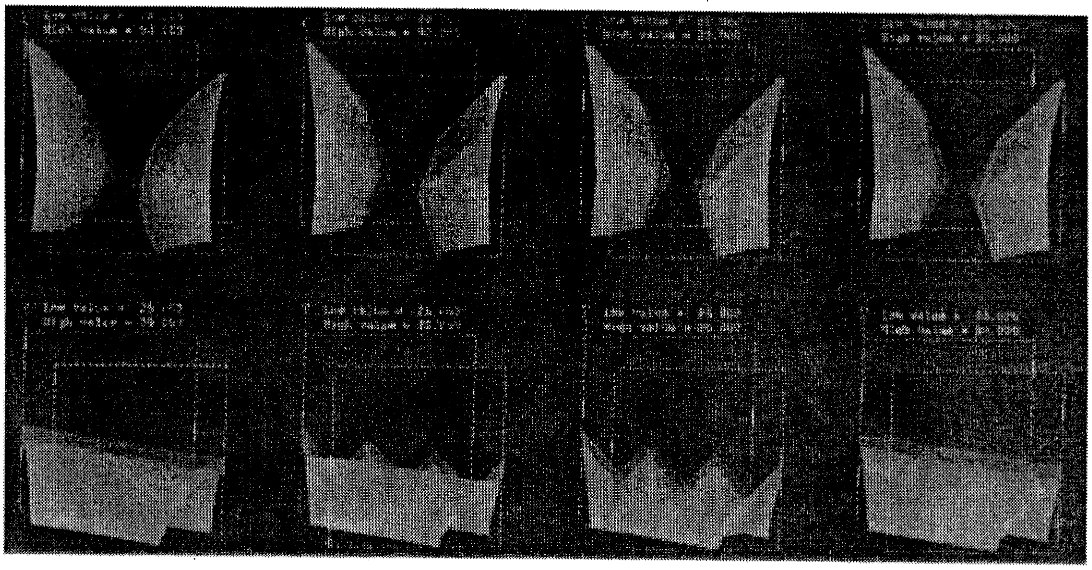

Figure 10: Comparison of disambiguation schemes for analytic functions f_1(upper) and f_2(lower). (From left to right) Control case(ideal); calculation without disambiguation; FAV & AD; GCH(CPG & QF).

Based on the above results, the most computationally-expensive QF algorithm is adopted herein as the reference disambiguation scheme for practical data sets, when the real field distribution is not available with any reconstruction scheme. Here, the HIPIP data set(64 × 64 × 64) shown in Fig.7(a) is reused to empirically analyze the topological consistency vs. temporal complexity trade-offs among the present auxiliary algorithms. The platform used for the experiment is an SGI Onyx system (CPU: R4400 × 2, Clock: 100MHz, RAM: 128MB, 2-way interleaved).

Table 1 compares the performance of four disambiguated MC solid fitting algorithms to extract $IV_D(127.5, 130.0)$ from the HIPIP data set in terms of the topological correctness and extraction time. The topological correctness is measured by counting the number of cubes where each disambiguation algorithm selects the same polyhedral block patterns as QF. Since the population of ambiguous cubes in the data set is very low(0.093%), there is little increase in overhead times to alleviate the topological ambiguity as the present auxiliary algorithms become more complex. As clearly seen from the table, however, the rate of correct block pattern selection is the lowest with the original MC with no disambiguation, and increases from FAV to QF.

[3] In the data sets, the field values are normalized onto one byte unsigned integer (0–255).

Table 1: Topological correctness/times in extracting HIPIP interval volume with disambiguated MC solid fitting algorithms.

Aux algorithm	Correctness	Extraction time
none	0.432	3.171
FAV	0.782	3.171
A D	0.799	3.176
CPG	0.910	3.179
Q F	1.000	3.183

[Time: in CPU seconds]

3.2 Acceleration with BONO

3.2.1 BONO definition and acceleration algorithm

Next, acceleration of the above-mentioned MC solid fitting algorithms is attempted using an existing hierarchical spatial index structure called *Branch-on-Need Octree*(BONO)[27].

BONO is a spatially-efficient octree for volumes whose resolutions are not of a power of two, and each node of which holds a min-max pair of field values of voxels in the corresponding subvolume. Fig.11 shows a 2D example of 5 × 6 volume data and the corresponding BONO structure.

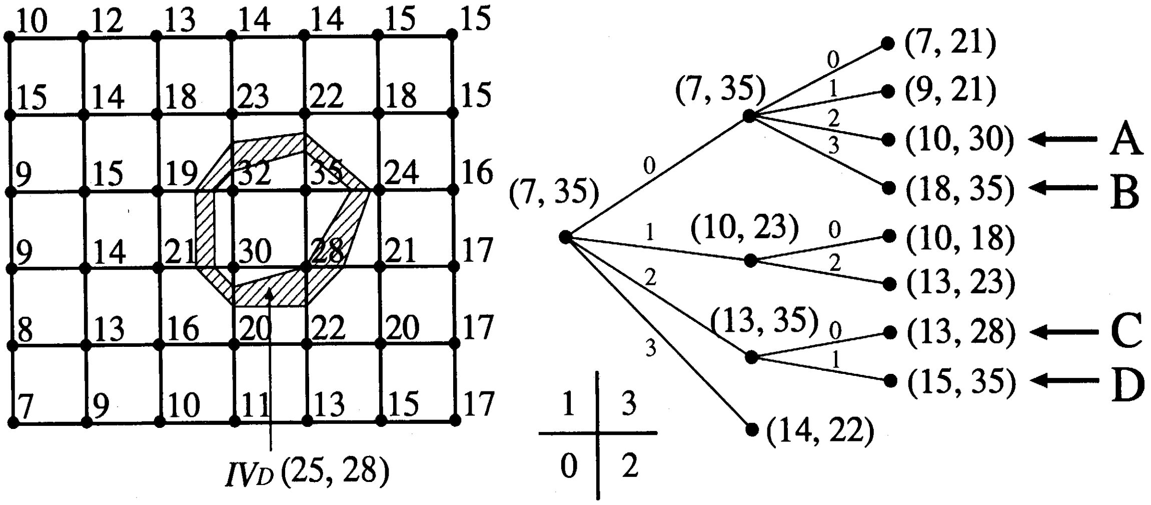

Figure 11: Volume data example and its BONO structure(2D).

BONO was originally devised for making the MC isosurfacing algorithm faster. Once a target field value has been given, the recursive traversal of BONO with the value makes it possible to avoid relatively large regions not intersecting the desired isosurface. Even if the target field values are changed during interactive exploration, the original BONO can be used for the volume. Hence, the overhead of BONO creation prior to repeated isosurfacing is generally acceptable.

The acceleration principle can be extended to the extraction of α and β-cubes in a straight-forward manner. Fig.12 is the outline of an accelerated version of the solid fitting algorithm[6].

For the 2D example in Fig.11, a test is performed for the accelerated solid fitting algorithm in Fig.12 to extract $IV_D(25, 28)$ from the volumetric data set. The procedure is started simply by calling BONO_traverse(root), and the BONO is recursively traversed in the depth-first way with the min-max test at each node. Since the root has the node value (7, 35), case 5 is first selected to traverse the

```
procedure BONO_traverse(node)
  begin
    $I_{max}$ ← node.max; $I_{min}$ ← node.min;
    cube_case ← check_cube($I_{min}, I_{max}$);
    switch (cube_case)
      begin
        case 1 : break;
        case 2 : if bound(node) then
                   if not leaf(node) then
                     for each child c of node do BONO_traverse(c);
                   else extract IV from subvolume(node);
                 else break;
        case 3 : if not leaf(node) then
                   for each child c of node do BONO_traverse(c);
                 else extract $\alpha$ block from subvolume(node);
                 break;
        case 4 : if not leaf(node) then
                   for each child c of node do BONO_traverse(c);
                 else extract $\beta$ block from subvolume(node);
                 break;
        case 5 : if not leaf(node) then
                   for each child c of node do BONO_traverse(c);
                 else extract IV from subvolume(node);
                 break
      end
  end;
```

cube_case:

case	Order relationship among α, β, I_{min}, I_{max}	Interpretation of subvolume
1	$I_{max} < \alpha$ or $\beta < I_{min}$	outside interval volume
2	$\alpha < I_{min} < I_{max} < \beta$	inside interval volume
3	$I_{min} \leq \alpha \leq I_{max} \leq \beta$	equivalent to $IV_D(f_{min}, \beta)$
4	$\alpha \leq I_{min} \leq \beta \leq I_{max}$	equivalent to $IV_D(\alpha, f_{max})$
5	$I_{min} < \alpha < \beta < I_{max}$	otherwise

Figure 12: A BONO-based accelerated algorithm for solid fitting.

subtree whose root is the child node 0 representing the lowerleft 4 × 4 quadrant, and so on. Finally, only the 16 cubes(53%) covering $IV_D(25, 28)$ corresponding to the four leaves from A to D are processed by the algorithm(Fig.11).

The interval volume module's control panel has the BONO creation toggle at the bottom(Fig.7(a)). If the toggle is turned on, the corresponding BONO is created only when a data set is input to the module for the first time, and the BONO continues to be referred to through subsequent execution of the network with the same data set.

3.2.2 Experiments

As a more practical case, the 3D electron density distribution around an H_2 molecule in Fig.1 is chosen. The original 3D continuous density distribution was sampled to produce five volumetric data sets with different resolutions(from 32^3 to 128^3). The experiments were run using an SGI Indy system (CPU: R4000, Clock: 100MHz; RAM: 64MByte).

Table 2 summarizes the times/spaces related to the interval volume extraction, where *cube hit rate* represents the percentage of cubes with intersecting interval volume boundaries(patterns 1–14 in Fig.6). The

Table 2: Comparison of times/spaces of MC solid fitting with/without BONO in extracting H_2 interval volume in Fig.1. In the formatting phase, internal structure of polyhedral data is converted into AVS geometry type.

#cubes	Phase	Without BONO	With BONO			
		A	B	B / A	# nodes	Cube hit rate
32^3	BONO Creation	—	0.10	—	4,681	8.62%
	IV extraction	0.76	0.37	48.68%		
	Formatting	0.16	0.16	100.00%		
	Total	0.92	0.63	68.48%		
48^3	BONO Creation	—	0.35	—	15,777	5.57%
	IV extraction	2.30	0.79	34.35%		
	Formatting	0.29	0.29	100%		
	Total	2.59	1.43	55.21%		
64^3	BONO Creation	—	0.80	—	37,449	4.15%
	IV extraction	4.99	1.60	32.06%		
	Formatting	0.51	0.51	100.00%		
	Total	5.50	2.91	52.91%		
96^3	BONO Creation	—	2.86	—	126,369	2.73%
	IV extraction	15.79	3.54	22.42%		
	Formatting	1.03	1.03	100.00%		
	Total	16.82	7.43	44.17%		
128^3	BONO Creation	—	6.50	—	299,593	2.03%
	IV extraction	36.15	7.64	21.13%		
	Formatting	1.72	1.72	100%		
	Total	37.87	15.86	41.88%		

[A, B: in CPU Seconds]

MC solid fitting algorithm used in the experiments is disambiguated with the AD scheme(Subsect.3.1). Note that no cacheing mechanism suggested in [27] for further acceleration is employed here. As the data sets get bigger, the cube hit rates decrease, and thus making the times for finding intersecting cubes in BONO smaller. Also, in all the cases, BONO creation times are insignificant. If those times are neglected, interval volumes can be extracted roughly three or four times faster with BONO than without BONO.

From other experiments in [6], it was revealed that the same degree of acceleration is realized with the sorted cube list-based algorithms[8] as well. However, from the viewpoint of spatial/temporal complexities to create additional index structures(BONO/sorted cube list), the present BONO-based algorithm is judged to be more efficient. For more detailed discussions, see [6].

Next, a time-evolving volumetric data set(100 frames of 60 × 60 × 60 voxels) representing the process of a simulated H^+-H collision shown in Fig.15[5] is used to investigate the sensitivity of the BONO-based acceleration to the spatial coherence of the 4D volumetric data sets. Fig.13 plots accumulatively as a function of frame IDs, the transition of computation times for several processing phases, including BONO creation, polyhedral block extraction, and calculation of statistics of interval volume[6]. The left graph is for the case without BONO, and the right one for the case with BONO. Vertical dotted lines

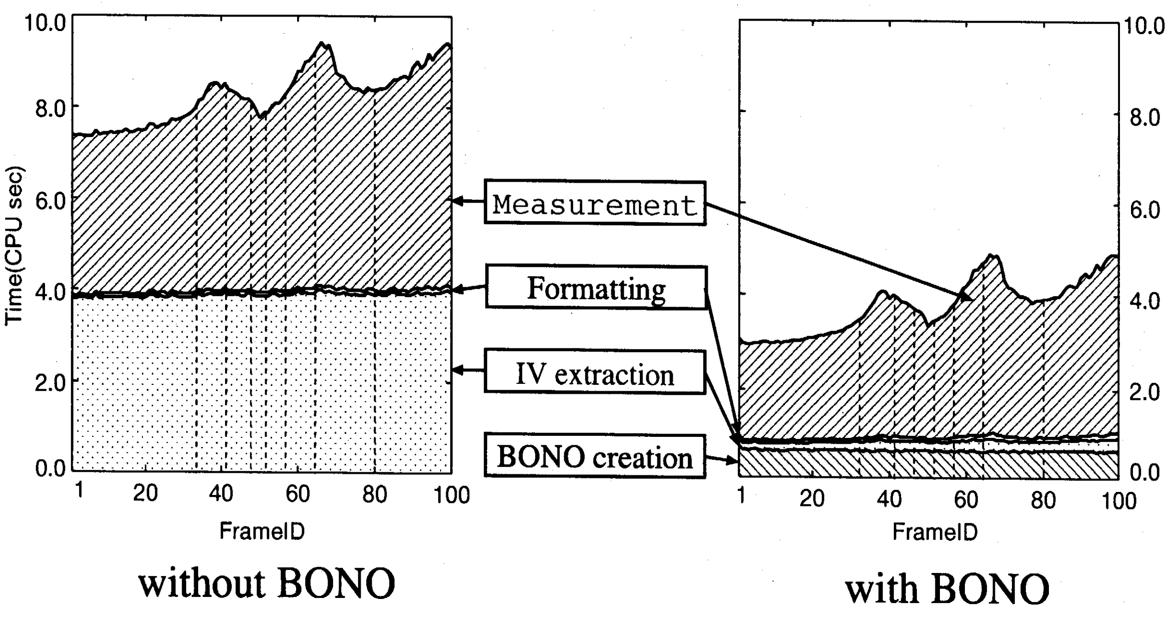

Figure 13: Sensitivity of the solid fitting algorithms with/without BONO to the spatial coherence of the volume data sets shown in Fig.15.

appearing in the both graphs denote the timings corresponding to the snapshots shown in Fig.15.

With the benefit from BONO, the extraction times are reduced to about one-fourth on average as the times without BONO. In fact, BONOs created through the frame sequence diminish the number ratio of the cubes processed by the modified MC solid fitting algorithm to 0.310 on average(SD = 0.156). Note that although the relative overhead is heavier than the other processing phases, the averaged time for interval volume measurement of 4.26 seconds(SD = 5.53) without BONO is also reduced to 2.90 seconds(SD = 5.57) with BONO. It is concluded that BONO can serve as an effective spatial index structure that has the potential to realize high and uniform frame rates when extracting a time series of interval volumes.

4 Interval Volume Operations

4.1 Focusing

One of useful interval volume-related operations is *focusing*, which controls the location and length of the interval to investigate properties such as the magnitude of errors hidden in a given volumetric data set, and to search for a target isosurface from the volume. When errors in the field value of a given volume are relatively large, it suffices to obtain an interval volume with an appropriate length of interval (Recall the example in Fig.2). Fig.14 illustrates an example of the focusing process on the H_2 volume of Fig.1, where the characteristic electron density value which splits the single isosurface into two parts around the hydrogen nuclei is sought. Focusing is expected to provide a powerful navigation mechanism for the topological analysis of interval volume. If several statistics over the ROI are derived from the surface area, total volume, and field integral during the focusing process(see Subsect.4.2), they can provide useful information for making the focusing process faster.

4.2 Measurement-coupled visualization

IV_D is a geometric model which is also suitable for *measurement-coupled visualization* of volumetric ROIs[5]. In addition to the capability of the surface area, total volume (displayed as solid measure in the measurement page), and field integral of interval volumes(Fig.7(b)), the interval volume module can solve a reverse problem of the conventional style of visualization. That is, it employs the bisection method

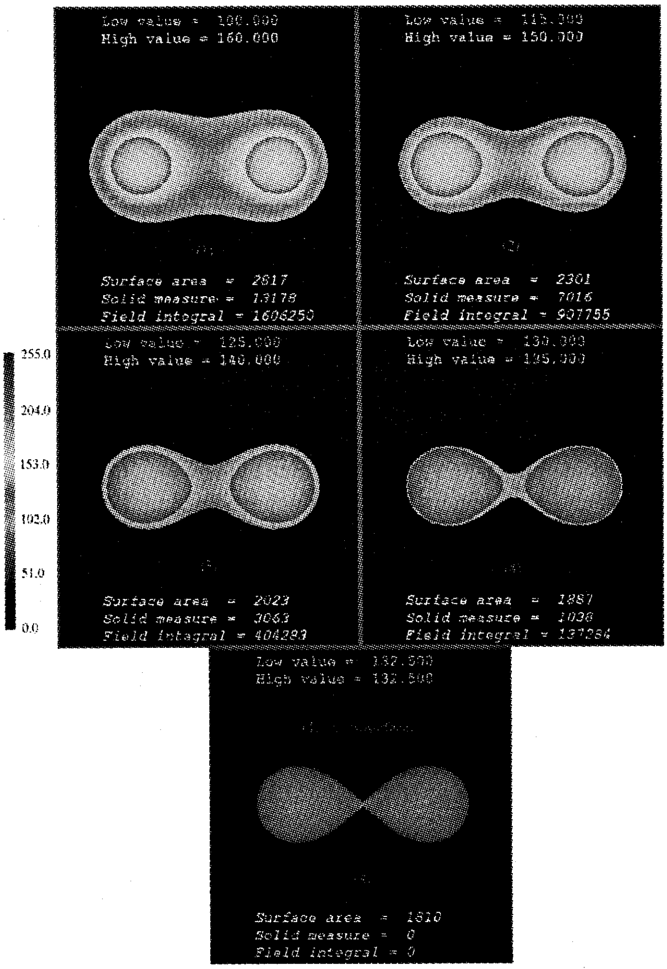

Figure 14: Example of focusing process.

to automatically find an interval limit value α (β) to meet the following integral equations for a given pair of β_0 (α_0) and I_0:

$$\int\int\int_{IV(\alpha,\beta_0)} f(x,y,z)dxdydz \quad = \quad I_0 \qquad (3)$$

$$\int\int\int_{IV(\alpha_0,\beta)} f(x,y,z)dxdydz \quad = \quad I_0 \qquad (4)$$

The interval volume module provides this calculation function through the auto-integration page(Fig.7(b)).

Consider, for example, the visualization of H^+–H(proton to hydrogen atom) collision simulation[4, 5]. In order to understand the collision dynamics, it is necessary to illustrate how the so-called "electron cloud" becomes distorted as the incident proton gets close to the target hydrogen atom. To do this, for each simulated time-step, a solid region with a constant existence probability of an electron has to be identified through the derivation from a simulated 4D data set for the time-evolving, electron density distribution.

Fortunately, the existence probability of an electron is given as the integration $P(\alpha, \rho_{max})$ of electron density ρ over the subvolume specified with $IV(\alpha, \rho_{max})$:

$$P(\alpha,\rho_{max}) \quad = \quad \int\int\int_{IV(\alpha,\rho_{max})} \rho(x,y,z)dxdydz, \qquad (5)$$

Since Eq.(5) is a special case of Eq.(3), if P_0 is given, the lower limit α such that $P(\alpha, \rho_{max}) = P_0$ can be automatically found.

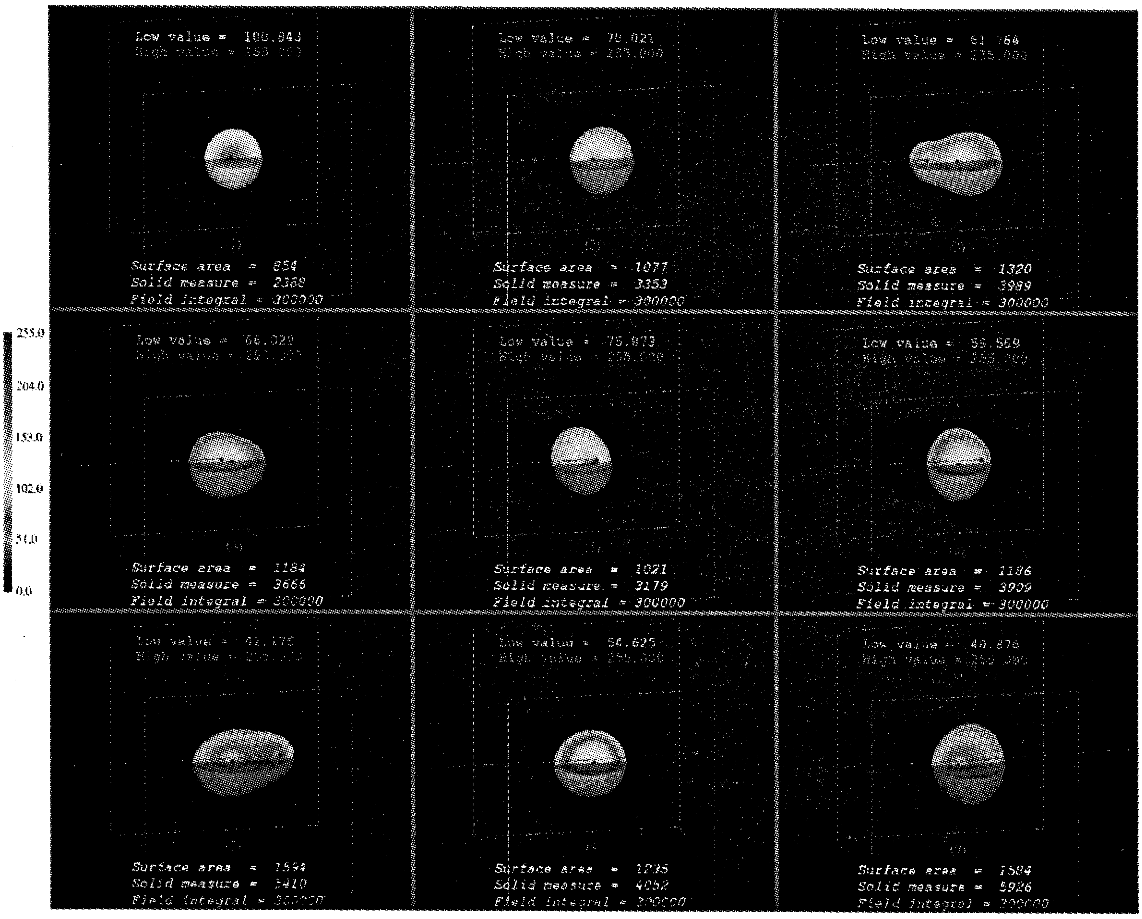

Figure 15: Snapshots excerpted from H^+-H collision animation using interval volume with constant electron existence probability.

Fig.15 shows nine snapshots excerpted from an animation of a time-evolving electron cloud, depicted with interval volumes of a constant P_0. Cutting away the upper-front quadrant of the visualization domain allows for direct observation of the color-coded change in existence probability of an electron within the electron cloud, as well as the relative position of the hydrogen nucleus(blue) and trajectories of the marching proton(red). For more detailed discussions, see [5].

4.3 Spatial coherence of volumetric data sets

Spatial coherence of a volumetric data set is the state in which associated field values do not drastically change in adjacent voxels[5]. The coherence of a volumetric data set is expected to have a strong influence on the spatial and temporal complexities of geometric structures to be extracted from the data set. Therefore, estimating the degree of coherence of a given data set prior to geometric feature extraction plays a key role in realizing a *time-critical* environment for *indirect* volume visualization. Herein, in order to measure the volumetric coherence, well-known, second-order grey-level statistics to measure 2D textures[19] are extended to 3D[22].

An estimate of the degree of volumetric coherence can be obtained by examining how often the possible pairs of field values in a given, regular, volumetric data set occur in the set of relative positions.

Let $\delta \equiv (\theta, \phi, r)$ be a 3D displacement defined in Fig.16(a). Let P_δ be a *co-occurrence matrix* whose (i, j) element represents the frequency that a voxel (x_0, y_0, z_0) having field value i occurs in position δ, relative to a voxel (x_1, y_1, z_1) having field value j $(f_{min} \leq i, j \leq f_{max})$. Then, let P be the average of P_δ's for a set of displacement δ's of a given size r in various direction. Here, 26 neighbors around a voxel(the Hamming distance $r = 1$) are examined for P to obtain isotropic information about the spatial distribution of field values in a volumetric data set (Fig.16(b)).

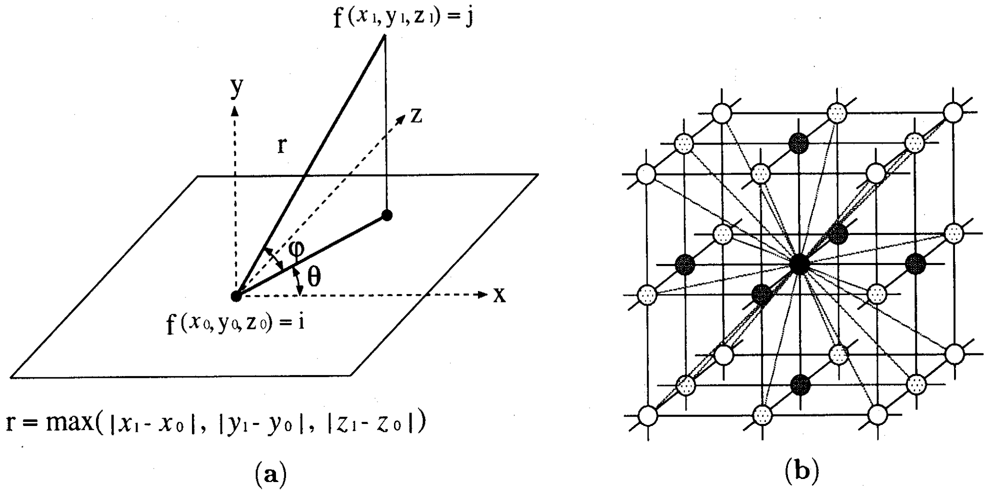

Figure 16: Settings for volumetric coherence analysis. (**a**) Definitions of displacement δ; and (**b**) possible directions for volume statistics.

As a candidate for the *Volumetric Coherence Measure*, the following statistic is derived from the averaged co-occurrence matrix P[22]:

$$VCM \equiv - \sum_{i=0}^{n_l-1} \sum_{j=0}^{n_l-1} \frac{(i-j)^2}{{n_l}^2} P(i,j) \log_{10} P(i,j)$$

where n_l denotes the number of possible integer field values ($= f_{max} - f_{min} + 1$).

The statistic VCM can indicate the degree of volumetric coherence from the viewpoint of gray-scale uniformity of field values, since the two types of spatial distribution expressiveness of Haralick's 2D statistics for textures, that is, *contrast* and *entropy*[19], are retained. Obviously, the more coherent a given data set becomes, the smaller the VCM value becomes. For comparison, the current interval volume module provides the user with the measurement capability with 3D versions of contrast and entropy as well as VCM(see the coherence page in Fig.7(b)).

In [22], the feasibility of the coherence measure VCM is illustrated with a collection of the Chapel Hill's testbed volumetric data sets[16]. It is clarified from thorough examination that VCM is correlated strongly to:

- Averaged cube hit rate (percentage of cubes minimally covering the location of interval volumes),
- Averaged number of interval volume boundary patches per cube,
- Percentage of ambiguous cube faces, and
- Percentage of the topological correctness of interval volume polyhedral block selections with the disambiguation schemes of Subsect.3.1.

This result may be used as a useful guideline to resolve the topological consistency vs. temporal complexity trade-offs among the disambiguation schemes.

Hereafter, another effectiveness of the VCM in a sort of multi-scalar visualization is focused on. In general, through the course of R&D, scientists and engineers often investigate mutual relationships among different sources of information by combining multiple sets of the related data into a single image, which exposes regions that are either positively or negatively correlated[24]. A commonly used visualization technique to accomplish this is extraction of geometric structures from one scalar field, which are colorized with another field. It is well known that little correlation information can be obtained from disordered geometric structures which are extracted from a scalar field with high complexity (low coherence). Thus, one idea is to use the VCM to measure the coherence in advance for each scalar field in a given pair, and to automatically choose a more coherent field for geometric structure extraction and

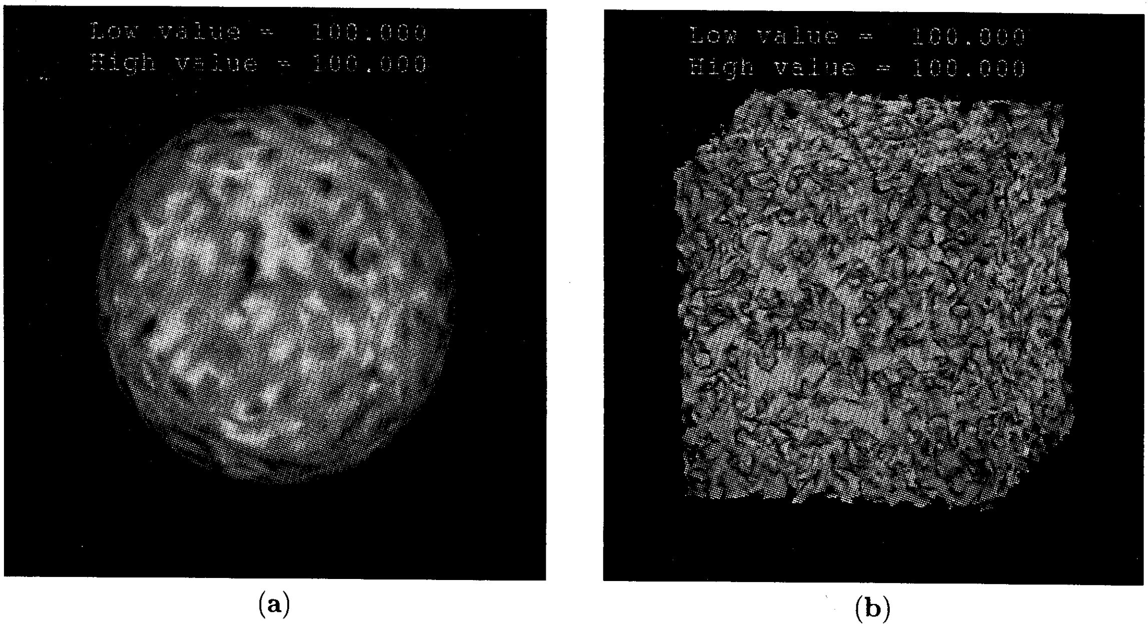

Figure 17: Coherence-dependent colored isosurfacing of two volumes. (**a**) Mapping Gaussian and noise volumes to shape and color, respectively; and (**b**) mapping Gaussian and noise volumes to color and shape, respectively.

to use the other field for the structure colorization. This function is available from the current interval volume module via the shape and color page(Fig.7(b)).

First, a pair of concentrically distributed density volume with a Gaussian function and a 3D white noise volume with the Perlin's solid texturing function[17] is assumed to be given. The size of the two volumetric data sets is commonly 32 x 32 x 32. Fig.17 shows two possible mappings for *colored isosurface*[4]. Clearly, the mapping (a) conveys more intuitive and comprehensible information to the viewers when the actual characteristics of the volumetric fields are unavailable a priori. In fact, the VCM values for the Gaussian volume and the noise volume are 4.71×10^{-3} and 2.00×10^{0}, respectively. From these results, the mapping (a) is judged to be more desirable.

Next, as a more practical example, ***bioconvection data visualization***[7] is adopted. ***Bioconvection***, named by Platt[18], forms characteristic aggregation patterns, like fingers, beneath the surface of cultures of aquatic microorganisms(Fig.18). In the two decades, several researchers have attempted to simulate the phenomena numerically to investigate the underlying mechanisms[10]. Since the interaction of microorganisms with surrounding water is to be considered, the main goal of the bioconvection simulation is to correlate distribution of microorganisms with the structure of the fluid flow field. Suppose that a given multi-scalar data set available here represents the density of microorganisms and the magnitude of velocity of water in which microorganisms are present.

The interval volume module has two import ports for volumetric data sets, and calculates the VCM for each of the imported data sets to automatically alternate the shape&color mappings with each other(Fig.7(b)). Fig.19 shows the visualization result of the data set using an interval volume-centered module network. Since the microorganism density filed is more coherent than the velocity magnitude field in terms of VCM (6.42×10^{-2} to 3.56×10^{-1}), the module network automatically produces an iso-microorganism density surface (degenerated interval volume) on which color-coded velocity magnitude is displayed. The visualization result reveals a certain number of cyclic convective flow patterns that quantitatively resemble the real ones in Fig.18.

[4] Isosurface $IS(\alpha)$ is nothing less than the degenerated interval volume $IV(\alpha, \alpha)$.

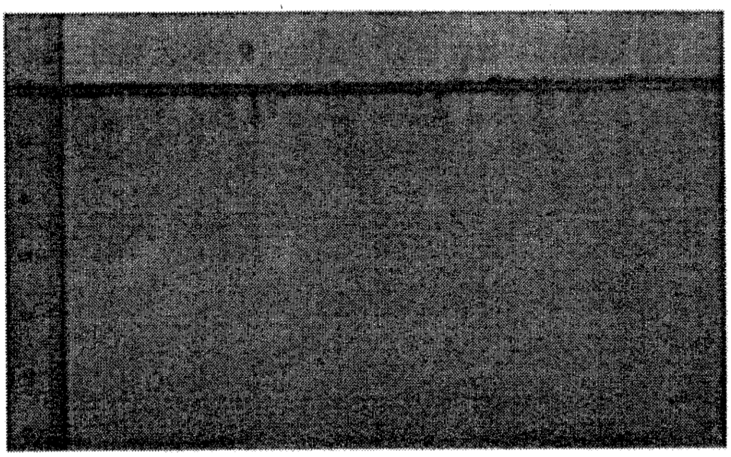

Figure 18: Real bioconvective patterns. Side view of finger-like aggregation patterns of a flagellate *Heterosigma Akashiwo*. Courtesy of Akira Harashima, The National Institute of Environmental Studies, Japan.

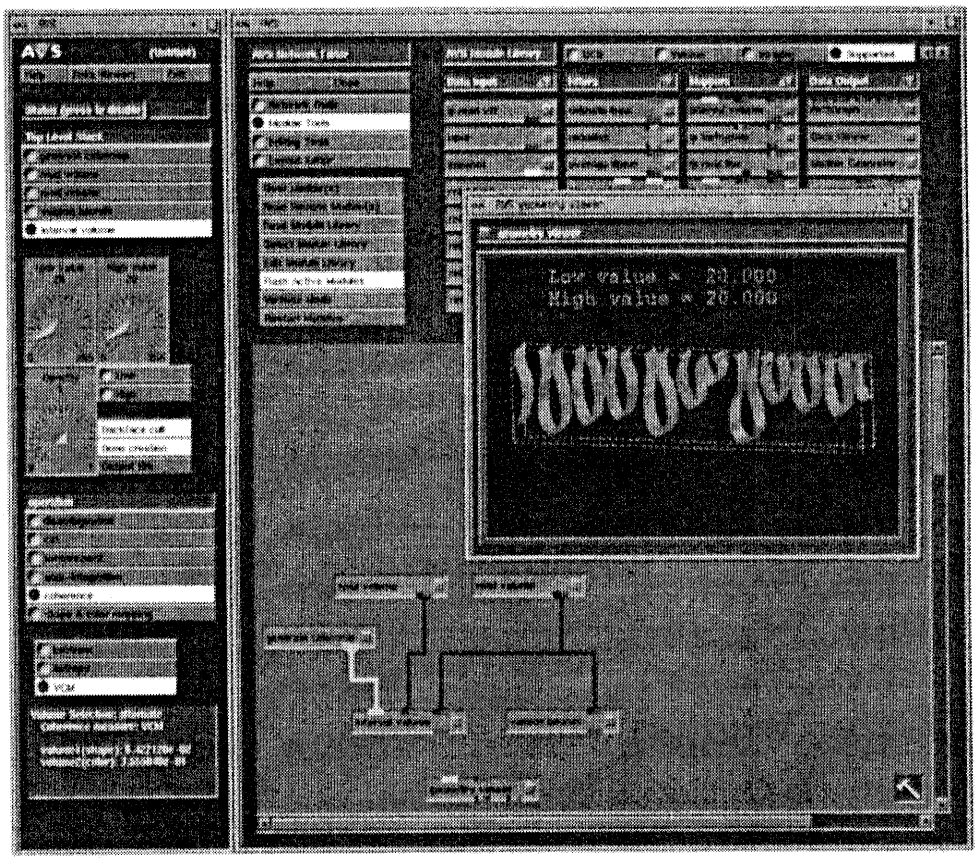

Figure 19: A module network for adaptive colored solid fitting and visualization results.

5 Conclusions

This paper demonstrates the solid fitting concept embodied with a geometric data model, called interval volume, as an effective tool for volumetric field interval analysis. As described in Sect.1, solid fitting inherits several advantages from both SF and DVR. However, these traditional volume visualization approaches still have their own merits to be used in practical situations. Rather, it is a challenging theme to incorporate the concept of solid fitting into a hybrid volume visualization environment, where each approach continues to play a complementary role with one another(Fig.20).

The advance use of interval volume is viewed as a significant step towards effectively inducing the exploration capability of the traditional approaches. For example, interval focusing can search for a set of critical target field values for isosurfaces to manifest themselves as an efficient interface for volumetric data comprehension. On the other hand, efficient field interval analysis with interval volume also makes it possible for one to decide the optimal color/opacity transfer functions and viewing-related parameter values for DVR, to produce final high-quality images and convey the most meaningful aspects of the target object. Interval volume can also be used as a *bounding* data structure for selective and fast DVR as well. Furthermore, the measurement capability, including the coherence estimation, leads to more sophisticated visualization methodologies that are indispensable for a deeper understanding of the target objects.

The present interval volume module serves as a versatile module used in a new generation modular visualization environment, called *GADGET*[5][7]. GADGET provides a knowledge base-assisted guidance mechanism so as to allow one to specify visualization goals and accuracy/complexity requirements, and to interactively design necessary application module networks.

Extensions of all the discussions on solid fitting techniques to *unstructured* case are being undertaken for visualizing large-scale finite element analysis problems[21].

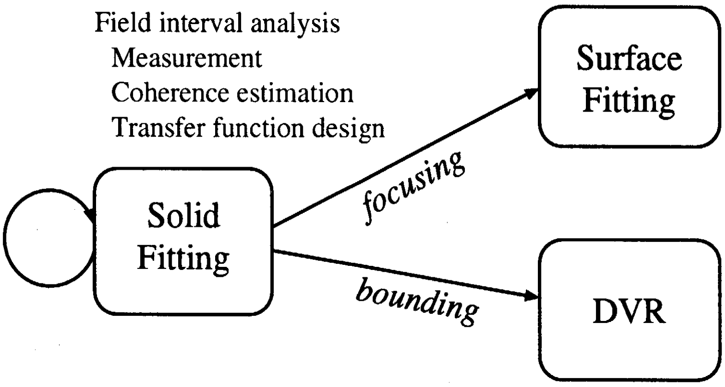

Figure 20: A hybrid volume exploration environment.

Acknowledgements

This paper is based on the presentation at the Dagstuhl Seminar on Scientific Visualization held in June 1997. The authors would like to express their gratitude to Arie Kaufman, Gregory M. Nielson and Hans Hagen, for giving a chance to present this work in the seminar. The Telecommunication Advancement Foundation, Tokyo provided Fujishiro, one of the co-authors, with financial support in participating in the seminar. Thanks are due to Yuichi Ohta at University of Tsukuba, who provided them with a valuable suggestion to use the statistical texture descriptions for developing the *VCM*. The authors have been benefiting also from continuous discussions with Hiroshi Sato of Ochanomizu University and Xiaoyang Mao of Yamanashi University. Finally, the authors appreciate comments on the English from Karen Vierow of Nuclear Power Engineering Corporation.

[5] Goal-oriented Application Design Guidance for modular visualization EnvironmenTs.

References

[1] Advanced Visual Systems Inc.: *AVS Developer's Guide*, 1992.

[2] Cameron, G.(ed.): "Special Focus: Modular Visualization Environments (MVEs)," *Computer Graphics*, vol. 29, no. 2, pp. 3–60, May 1995.

[3] Crawfis, R. A.: "Real-Time Slicing of Data Space," in *Proc. IEEE Visualization '96*, Sun Francisco, ACM Press, New York, October–November 1996, pp. 271–277.

[4] Fujishiro, I., Maeda, Y. and Sato, H.: "Interval Volume: A Solid Fitting Technique for Volumetric Data Display and Analysis," in *Proc. IEEE Visualization '95*, Atlanta, IEEE Computer Society Press, Los Alamitos, October–November 1995, pp. 151–158.

[5] Fujishiro, I., Maeda, Y., Sato, H. and Takeshima, Y.: "Volumetric Data Exploration Using Interval Volume," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 2, pp. 144–155, June 1996.

[6] Fujishiro, I., Takeshima, Y., Maeda, Y.: "A Fast Interval Volume Extraction Algorithm Using a Hierarchical Volumetric Spatial Index Structure" (in Japanese), *Transactions on Information Processing*, Information Processing Society of Japan, vol. 38, no. 8, pp. 1593–1602, August 1997.

[7] Fujishiro, I., Takeshima, Y., Ichikawa, Y. and Nakamura, K.: "GADGET: Goal-Oriented Application Design Guidance for Modular Visualization Environments," in *Proc. IEEE Visualization '97*, Phoenix, ACM Press, New York, October 1997, pp. 245–252, 548.

[8] Giles, M. and Haimes, R.: "Advanced Interactive Visualization for CFD," *Computing System in Engineering*, vol. 1, no. 1, pp. 51–62, 1990.

[9] Guo, B.: "Interval Set: A Volume Rendering Technique Generalizing Isosurface Extraction," in *Proc. IEEE Visualization '95*, Atlanta, IEEE Computer Society Press, Los Alamitos, October–November 1995, pp. 3–10.

[10] Harashima, A., Watanabe, M. and Fujishiro, I.: "Evolution of Bioconvection Patterns in a Culture of Motile Flagellates," *The Physics of Fluids*, vol. 31, no. 4, pp. 764–775, April 1988.

[11] Kaufman, A. E.(ed.): *Volume Visualization*, IEEE Computer Society Press, 1990.

[12] Lorensen, W. E. and Cline, H. E.: "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *ACM Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, vol. 21, no. 4, pp. 163–169, July 1987.

[13] Mäntylä, M.: *An Introduction to Solid Modeling*, Chap. 6, Computer Science Press, Rockville, 1988.

[14] Nielson, G. M. and Hamann, B.: "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes," in *Proc. IEEE Visualization '91*, Phoenix, IEEE Computer Society Press, Los Alamitos, October 1991, pp. 83–90.

[15] Nielson, G. M. and Sung, J.: "Interval Volume Tetrahedrization," in *Proc. IEEE Visualization '97*, Phoenix, ACM Press, New York, October 1997, pp. 221–228.

[16] Noodleman, L., Case,D. University of North Carolina, Chapel Hill.
ftp://ftp.cs.unc.edu/pub/projects/softlab/CHVRTD/volI

[17] Perlin, K.: "An Image Synthesizer," *ACM Computer Graphics* (*SIGGRAPH '85 Conference Proceedings*), vol. 19, no. 3, pp. 287–296, July 1985.

[18] Platt, J. R.: "Bioconvection Patterns in Cultures of Free-Swimming Microorganisms," *Science*, vol. 133, pp. 1766–1767, 1961.

[19] Rosenfeld, A. and Kak, A. C.: *Digital Picture Processing*, 2nd Ed., Vol. II, Ch. 12, Academic Press, Orlando, 1982.

[20] Takeshima, Y., Fujishiro, I. and Yamashita, Y.: "Alleviating Topological Ambiguities in Interval Volume Extraction" (in Japanese), *SIGNotes* 97-CG-84-5, Information Processing Society of Japan, February 1997, pp. 25–32.

[21] Takeshima, Y., Fujishiro, I.: (1998) "Unstructured Volume Visualization: Approaches and Issues," in *the Conference on Computational Enginnering and Science*, Tokyo, The Japan Society for Computational Engineering and Science, May 1998, vol. 3, no. 1, pp. 103–106.

[22] Takeshima, Y. and Fujishiro, I.: "Measuring Volumetric Coherence," in *ACM SIGGRAPH98 Conference Abstracts and Applications*, Orlando, July 1998, p. 260.

[23] Udupa, J. K. and Odhner, D.: "Shell Rendering," *IEEE Computer Graphics and Applications*, vol. 13, no. 6, pp. 58–67, November 1993.

[24] Upson, C.: "Volumetric Visualization Techniques," in *State of the Art in Computer Graphics – Visualization and Modeling* – (Rogers, D. F. and Earnshaw, R. A.(eds.)), Springer-Verlag, New York, 1991, pp. 313–350.

[25] Van Gelder, A. and Wilhelms, J.: "Topological Considerations in Isosurface Generation," *ACM Transactions on Graphics*, vol. 13, no. 4, pp. 337–375, October 1994, (also extended abstract available as ACM Computer Graphics, vol. 24, no. 5, pp. 79-86, November 1990).

[26] Wang, S.: "The Problem of the Normal Hydrogen Molecule in the New Quantum Mechanics," *Physical Review*, vol. 31, pp. 579–586, April 1928.

[27] Wilhelms, J. and Van Gelder, A.: "Octrees for Faster Isosurface Generation," *ACM Transactions on Graphics*, vol. 11, no. 3, pp. 201–227, July 1992.

[28] Wyvill, B., McPheeters, C. and Wyvill, G.: "Data Structures for Soft Objects," *The Visual Computer*, vol. 2, no. 4, pp. 227–234, August 1986.