

# 階層的空間索引構造を用いた区間型ボリュームの高速抽出

藤代一成<sup>†</sup> 竹島由里子<sup>††</sup> 前田裕治<sup>†††</sup>

対象ボリュームデータの解析は、指定フィールド値やビューイングパラメータ値の変更を繰り返しながら進められることから、幾何学的特徴の抽出や表示の高速化が望まれている。ボリュームデータの容量は膨大であり、計算資源に大きな負担をかけるが、使用する補助データ構造や走査方法を工夫することによって、関連するアルゴリズムの時間効率を大きく改善することができる。筆者らは先に、効果的なボリュームデータの表示や解析を目的として、フィールド値がある有限閉区間に属するような三次元サブボリュームをソリッドデータとして幾何学的に表現する「区間型ボリューム」を提案した。本論文では、この区間型ボリュームを、階層的ボリューム索引構造として知られる Branch-on-Need Octree を用いて、高速に抽出するアルゴリズムを提案する。さらに、商用データビジュアライザ AVS Ver.5 上に本法を実装し、陽子・水素原子衝突などの事例を用いて、既存の手法との効率の比較を行い、その有効性を示す。

## A Fast Interval Volume Extraction Algorithm Using a Hierarchical Volumetric Spatial Index Structure

ISSEI FUJISHIRO,<sup>†</sup> YURIKO TAKESHIMA<sup>††</sup> and YUJI MAEDA<sup>†††</sup>

Geometric features of a given volume dataset are expected to be extracted and rendered at interactive rate, because the dataset has to be analyzed repeatedly with different target field values and viewing parameter values. Although a volumetric dataset taxes the computer resources, the performance of related algorithms may be improved by judicious choice of auxiliary data structures and their appropriate traversal methods. For effective display and analysis of volumetric datasets, we have proposed a new type of geometric model, named *interval volume*, which represents a 3D subvolume with field values belonging to a closed interval. This paper represents an accelerated interval volume extraction algorithm using a hierarchical volumetric spatial index structure called Branch-on-Need Octree. The algorithm is implemented as an in-house module on a commercially-available visualization software AVS Ver.5. Several applications such as proton-hydrogen atom collision are used to prove the temporal efficiency of the presented extraction algorithm compared with other existing ones.

### 1. はじめに

ボリュームビジュアライゼーション (Volume visualization) は、多変量の三次元分布データであるボリュームデータの複雑な内部構造や動的振舞いを可視化する技術であり、対象の効果的な解析のためにさまざまな可視化手法が研究・開発されてきている<sup>1),2)</sup>。そのアプローチの1つに、与えられたボリュームデータからある基準を満たす幾何学的な特徴をサーフェスとして

抽出・表示する、サーフェスフィッティング (Surface Fitting; SF) がある。SF アプローチは、ボリューム全体を表示するわけではないので、ユーザは表示されていない箇所に留意して対象を解析する必要がある (図 1 (a) 参照)。しかし、幾何学的な中間データを生成せずにボリュームデータ全体の半透明な画像を生成するダイレクトボリュームレンダリング (Direct Volume Rendering; DVR) アプローチ (図 1 (c) 参照) と比較して、一般に高速な表示や操作が可能であり、空間コヒーレンスが高い場合には、データ格納のための記憶容量も節約できるなどの利点を持つ<sup>1),2)</sup>。

我々は先に、新しいボリューム幾何モデルである区間型ボリューム (Interval volume) を提案した<sup>3)~6)</sup>。これは、フィールド値がある有限閉区間に属するような三次元サブボリュームをソリッドとして表現したものであり、SF アプローチの代表的な手法の1つであ

<sup>†</sup> お茶の水女子大学理学部

Faculty of Science, Ochanomizu University

<sup>††</sup> お茶の水女子大学大学院人間文化研究科

Graduate School of Humanities and Sciences, Ochanomizu University

<sup>†††</sup> 筑波大学大学院工学研究科

Doctoral Program in Engineering, University of Tsukuba

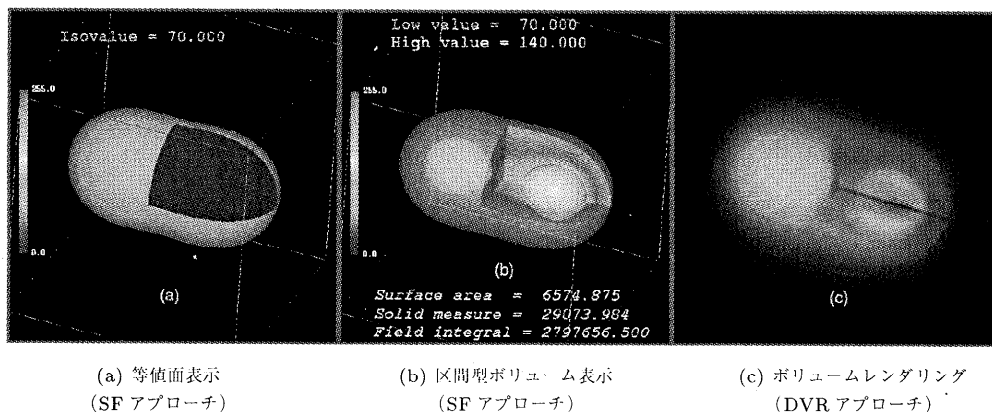


図1 水素分子電荷密度の三次元分布を可視化する3つの手法

Fig. 1 Three volume visualization methods for visualizing the stable 3D probability density distribution of an electron around an  $H^2$  molecule.

る、特定のフィールド値を持つボリュームの位置をパッチで連結する等値面 (Isosurface) の一般形を与えている (図1 (b) 参照)。この区間型ボリュームは、従来のSFアプローチと比較して、以下のような利点を持つ<sup>3)~6)</sup>。

- 区間型ボリュームは与えられたボリュームデータから幾何学的構造として抽出・表示されることから、DVRアプローチに近い直観的可視化が可能である一方、同アプローチほどの計算量を必要としない。よって、DVRアプローチで対象を詳細に可視化する前に、あらかじめ最適なビューイングパラメータ値や transfer function の定義などを決定する際に利用できる。すなわち、効果的なボリュームビジュアライゼーションのためのハイブリッドアプローチの形成に役立つ。
- 区間型ボリュームは、ソリッドデータとして抽出されることから、簡単な幾何学的計算によって形態的計測も可能であり、興味ある部位 (Region of Interest: ROI) の定量的解析を効率良く行える。
- 自然界の計測やシミュレーションによって得られたデータセットが、対象の持つ構造的あいまいさや、測定、数値モデル・数値計算による誤差を持つ場合、区間型ボリュームはフィールド値に一定の許容範囲 (Allowance) を与えることができるため、誤差を許容したボリュームスライス (厚みのある等値面) を ROI として指定することができる。また、区間の位置や幅を制御するフォーカシング操作<sup>3)~6)</sup>を用いることで、効果的に等値面化の目的フィールド値を探し出すことができる。

この区間型ボリュームのように、ボリューム全体ではなく、ある一定区間の興味あるフィールド値を持つ三

次元サブボリュームに着目して効率的に対象ボリュームを可視化する考え方は、Guo の区間集合 (Interval Set)<sup>7)</sup>、Crawfis のデータスペーススライシング<sup>8)</sup>など、最近の多くの研究例に採用されており、リアルタイムボリュームビジュアライゼーションへ向けての1つの技術的潮流を形成し始めていると考えられる。

SFアプローチを用いた対象ボリュームの解析は、指定フィールド値の変更を繰り返しながら進められる。これにともなう処理時間が長ければ長いほど、ユーザが対象に没入する可能性は低くなり、対象の探究にとって有益ではない。したがって、それらの操作にともなう幾何学的特徴の抽出や表示は高速に行われるべきである。しかし、ボリュームデータの容量は膨大であり、計算資源に大きな負担をかけている。そのため、DVRアプローチに比べ、SFアプローチが計算量的に有利であるとしても、実際の利用にあたっては十分であるとはいえない。そこで、使用する補助データ構造や走査方法を工夫することによって、アルゴリズムの効率を大きく改善することを試みることにした。

本論文では、区間型ボリューム抽出 (2章) を、ボリュームの情報を集約した階層的な空間索引構造である Wilhelms らの Branch-on-Need Octree (BONO)<sup>9)</sup> (3章) を用い、関心の少ない領域への無駄な走査を避けることにより高速化する<sup>6),10),11)</sup> (4章)。さらに、AVS<sup>☆</sup> Ver.5 上のモジュールとして、本論文で提案する高速化アルゴリズムを実装する。そして、キューブ内の最小フィールド値に基づいて、キューブへの索引をソートし、そのリストを用いる既存の高速等値面化アルゴリズム<sup>12)</sup> の拡張などとの比較や、陽子・水素

☆ AVSは米国 Advanced Visual Systems 社の商標である。

原子衝突の三次元電荷密度分布データの可視化実験から本手法の有効性を検証する(5章).最後に今後の課題をあげる(6章).

## 2. 区間型ボリューム

区間型ボリュームは,フィールド値がある有限閉区間に属するような三次元サブボリュームを,多面体ソリッドとして表現したものである.これは,区間の指定の仕方によってさまざまな意味を持つ.フィールド値全体の区間を指定すれば,区間型ボリュームはボリューム全体を表現し,区間の幅を0にとれば,区間型ボリュームは等値面に縮退する.また,フィールド値の周りに十分小さな値である誤差を考慮した領域を指定すれば,誤差を許容した等値面を表現することも可能である.このことから,区間型ボリュームは等値面の一般形を与えていると考えられる.本章では,この区間型ボリュームを定義し,その抽出方法について簡単に述べる.詳しくは,文献3)~6)を参照されたい.

### 2.1 区間型ボリュームの定義

$S$  を三次元ユークリッド空間内の矩形領域,  $d$  を有限閉区間  $[d_{\min}, d_{\max}] (C R) \rightarrow S$  を写像するフィールド関数とすると,区間型ボリューム  $IV(\alpha, \beta)$  は,次の式で表される:

$$IV(\alpha, \beta) = \{ (p, d(p)) \mid p \in S, \alpha \leq d(p) \leq \beta \}$$

ただし,  $d_{\min} \leq \alpha \leq \beta \leq d_{\max}$ .

与えられたボリューム  $V = (S, d)$  はつねに解析的に与えられるとは限らないので,  $V$  に各座標軸方向に等間隔な格子を割り当て,その格子点だけがフィールド値を持つような離散ボリューム  $V_D$  を考える.ここでは,この格子点をボクセル,各座標軸で隣接する8つのボクセルから成る立方体をキューブと呼ぶことにする.離散ボリュームでは,格子点以外のフィールド値を,格子点のフィールド値から trilinear 補間\*を用いて求める方法が広く用いられているので,ここでも採用する.

本論文では,与えられた離散ボリューム  $V_D$  から抽出される区間型離散ボリューム  $IV_D$  について考える. $IV_D$  は  $V_D$  のフィールド値を内部に持つ,多角形パッチの集合で表された多面体ソリッドとして抽出される.その多面体ソリッドの情報は,境界表現法 (Boundary representations: B-reps)<sup>13)</sup>を用いて表現する.以下では,本論文で扱うボリュームデータを,各ボクセル

が単一のスカラーフィールドを持つ正六面体格子構造のボクセルデータセットであると仮定するが,一般にはボクセルデータセットの構造を,曲線 (curvilinear) 格子まで緩和することが可能である.

### 2.2 区間型ボリュームの抽出

区間型ボリューム  $IV_D(\alpha, \beta)$  は,等値面生成アルゴリズムとして最もよく知られているマーチングキューブ (Marching Cubes: MC) 法<sup>14)</sup>を拡張することによって,離散ボリュームデータ  $V_D$  から抽出することができる<sup>3)~6)</sup>.拡張を行う際に基本となるMC法は,等値面パッチの構成に関する曖昧さを持つことが知られているが,ここでは, *Asymptotic Decider* 補助アルゴリズム<sup>15)</sup>によってMC法の曖昧さを緩和することにした<sup>6)</sup>.この補助アルゴリズムはキューブ内のフィールド値の trilinear 性の仮定の下では,キューブを再分割することなしに,正しい等値面パッチの位相選択が図れることが保証されている.なお,キューブ内におけるフィールド値の変化を高次の関数で近似する必要のある場合については,文献16)を参照していただきたい.

区間型ボリュームの抽出は,ソリッドの集合演算を活用する方法を用いて行われる.簡潔に述べると, $IV_D(\alpha, \beta)$  を生成するために,まず各キューブに対して  $IV_D(\alpha, d_{\max})$ ,  $IV_D(d_{\min}, \beta)$  を表す多面体ブロックを生成し,次にそれらのブロックの共通領域 (区間型ボリュームブロック) を特定する.以後,前者を  $\alpha$  キューブ,後者を  $\beta$  キューブと呼ぶことにする.そして,すべてのキューブに対して区間型ボリュームブロックを抽出した後,それらの和集合をとることにより区間型ボリュームを抽出する.このとき,各キューブ内のフィールド値は trilinear 性を保持しているため,隣接するキューブの共有面における区間型ボリュームブロックの多角形パッチは一致する.このことから,共有面の多角形パッチは省略可能となる\*\*.ここで,  $\alpha, \beta$  キューブに対する多面体ブロックパターンは,MC法のパッチパターンに,どちらの領域を選択するか向きを与えたものに過ぎないため,それぞれの個数は,オリジナルのMC法の多角形パッチパターンの個数に一致する.

この抽出方式を採用した理由は,キューブの各頂点を区間より下,区間内,区間より上の3値に分類し, $3^8 (= 6,561)$ 通りの組合せから区間型ボリュームの多面体ブロックパターンを直接的に求め,MC法と同様に回転・鏡像変換を用いて導かれる同相関係から同

\* キューブ内の任意の点  $P(x, y, z)$  のフィールド値  $f(x, y, z)$  を  $f(x, y, z) = c_0 + c_1x + c_2y + c_3z + c_4xy + c_5yz + c_6zx + c_7xyz$  によって近似する方式.係数  $c_0, \dots, c_7$  は,キューブの8つの頂点のフィールド値から一意に決定可能.

\*\*  $S$  の外向き境界は  $IV_D$  を閉じるために省略できない.

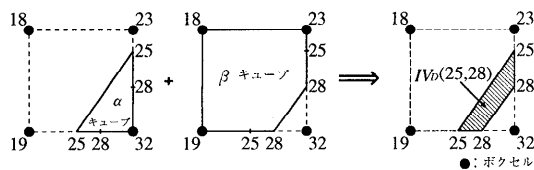


図2 単一キューブ内の区間型ボリューム抽出(二次元)  
Fig. 2 Interval volume extraction in a single cube (2D).

種類を構成したとしても、ルックアップテーブルに格納できる程度のパターン数にまで減少させられないと考えたからである。本方式に従えば、区間型ボリューム抽出アルゴリズムの計算量は、MC法の計算量である $O(N)$  ( $N$ は $V_D$ のキューブの総数)に等しい。

図2に、二次元における区間型ボリューム抽出の例を示す<sup>\*</sup>。図中の頂点の数字はフィールド値を表している。図のようなキューブから $IV_D(25, 28)$ を求めることを考える。キューブ内ではフィールド値のtrilinear性が仮定されていることから、フィールド値25の等値面は同図左のように現れ、実線で囲まれた領域が $\alpha$ キューブとなる。同様に $\beta$ キューブを求めると、真中のようになる。右の斜線部がこれらの共通領域となり、 $IV_D(25, 28)$ が求められる。

### 2.3 区間型ボリュームの計測

計測は、目的とする対象を総合的に理解するために不可欠な処理である。幸い、 $IV_D$ は、求めるサブボリュームの表面積や体積の計測に適した幾何学的モデルである<sup>3)~6)</sup>。実際、 $IV_D$ の表面積は、多角形パッチの表面積の総和をとることで近似計算できる。区間型ボリューム $IV_D(\alpha, \beta)$ 内のフィールドの積分値 $D(\alpha, \beta)$ は、区間型ボリュームが属するキューブを十分に細分し、サブキューブのボクセルにおけるフィールド値の総和として近似的に求められる。したがって、ROIのフィールド値の平均も導出可能であり、たとえば1章で述べたフォーカシング操作を高速に行うためのナビゲーション情報として利用することができる。

## 3. Branch-on-Need Octree

Wilhelmsらは、MC法による等値面抽出を高速に行うために、階層的なボリューム空間索引構造であるBranch-on-Need Octree (BONO)を提案した<sup>9)</sup>。本章では、与えられたボリュームに対して、どのようにBONOを作成するかを述べた後、BONOを用いて等値面を高速に抽出する方法を概説する。

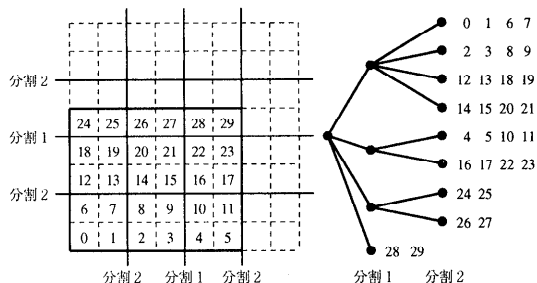


図3 二次元ボリュームデータの分割とBONOの構造  
Fig. 3 Volume data partition and BONO structure (2D).

### 3.1 BONOによるボリュームの分割

BONOでは、与えられたボリュームを階層的に分割し、作成したオクツリーの各ノードに、そのノードが示すサブボリューム内のフィールド値の最小値と最大値を格納しておく。

まず、与えられたボリュームに対して、各座標軸方向の解像度が2のべき乗からなるバウンディングボリュームを考え、各座標軸方向に再帰的に二等分していく。ここで、キューブ単位になるまで分割を行うと、オクツリーで表現したボリュームデータが、もとのボリュームデータよりも大きくなり、膨大な記憶領域を必要としてしまうため、ボリュームの分割は、各ノードにおけるサブボリューム内のキューブの個数が $2 \times 2 \times 2$ になるまでにとどめておく。

図3は、二次元における $5 \times 6$ のボリュームデータをBONOに基づいて分割した様子(左)と、対応するBONOの構造(右)を示したものである。ここで、与えられたボリュームのバウンディングボリュームは、 $8 \times 8$ のボリュームとなる。図中では実領域を実線、バウンディングボリュームを破線で表した。また、ボリュームの各キューブに与えた数字は、それぞれのキューブ番号を示している。BONOのリーフに割り当てたキューブ番号は、そのリーフが示すサブボリューム内のキューブの番号である。

上のよう分割を行い、作成されたBONOは、構造的には完全木であるが、すべての子ノードが空であるノードに対しては、空間を割り当てていない。つまり、そのノードが示すサブボリューム内に実領域が存在する場合にだけ、ノードを生成する。

### 3.2 BONOを用いた等値面の高速抽出

任意のフィールド値の等値面は、BONOを走査していき、等値面が存在するサブボリュームに効率的にアクセスすることによって抽出される。従来の等値面化では、与えられたボリュームのすべてのキューブにアクセスし等値面を生成していた。それに対し、BONO

<sup>\*</sup> なお、今後二次元においても三次元の用語を断りなく用いることにする。

を用いることで、不要なキューブへのアクセスを減少させることができるため、等値面を高速に抽出することが可能となる。ここで、与えられたボリュームに対して一度 BONO を作成しておけば、異なるフィールド値の等値面を求める場合においても、再利用可能であることに注意されたい。したがって、BONO の生成によるオーバーヘッドは、対話的ボリューム解析の環境下では無視できると考えられる (5 章参照)。

従来の MC 法と BONO を用いた MC 法による抽出時間を比較すると、与えられたボリュームに対する等値面が存在するキューブの割合が低いものほど高速に等値面が抽出できるという結果が得られている<sup>9)</sup>。前章で述べたように、区間型ボリュームは MC 法を拡張することによって抽出されるため、等値面の場合と同様に BONO を用いて無駄なキューブへのアクセスを減少させることによりその抽出を高速化できると考えられる。これが次章で提案する区間型ボリューム抽出の高速アルゴリズム開発の直接的動機である。

#### 4. BONO を用いた区間型ボリュームの高速抽出

本章では、区間型ボリュームを高速に抽出するアルゴリズムを提案する。また、このアルゴリズムを用いれば、区間型ボリュームが存在するキューブに効率的にアクセスすることで高速な計測も可能となる。

##### 4.1 区間型ボリューム高速抽出アルゴリズム

BONO を用いて、区間型ボリューム  $IV_D(\alpha, \beta)$  を求めることを考える。

まず、与えられたボリュームに対して、BONO を作成する。  $\alpha = \beta$  のとき、区間型ボリュームは等値面に縮退するため、  $\alpha < \beta$  であると仮定する。各ノードに与えられた、対応するサブボリューム内におけるフィールド値の最小値・最大値をそれぞれ  $I_{\min}, I_{\max}$  とすると、  $\alpha, \beta, I_{\min}, I_{\max}$  の大小関係は 6 通りである。それぞれの場合における、そのノードが示すサブボリュームと区間型ボリュームの関係は、表 1 の 5 ケースに分類できる。

case 1 の場合、そのノードが示すサブボリューム全体は求める区間型ボリューム外であり、また case 2 の場合も、サブボリューム全体が区間型ボリューム内であることは、各々のキューブ内でフィールド値の trilinear 性を仮定していることから明らかである。ただし、case 2 に属するキューブが求める区間型ボリュームブロックの境界に位置するならば、区間型ボリュームブロックを閉じるために、仮の境界として必要となる。case 3 の場合、サブボリューム全体が  $IV_D(d_{\min}, \beta)$  と一致するため、

表 1  $\alpha, \beta, I_{\min}, I_{\max}$  の大小関係とサブボリュームの解釈  
Table 1 Interpretation of subvolumes with respect to  $\alpha, \beta, I_{\min}$  and  $I_{\max}$ .

case	$\alpha, \beta, I_{\min}, I_{\max}$ の大小関係	サブボリュームの解釈
1	$I_{\max} < \alpha$ または $\beta < I_{\min}$	区間型ボリューム外
2	$\alpha < I_{\min} < I_{\max} < \beta$	区間型ボリューム内
3	$I_{\min} \leq \alpha \leq I_{\max} \leq \beta$	$IV_D(d_{\min}, \beta)$ と一致
4	$\alpha \leq I_{\min} \leq \beta \leq I_{\max}$	$IV_D(\alpha, d_{\max})$ と一致
5	$I_{\min} < \alpha < \beta < I_{\max}$	その他

それぞれのキューブの区間型ボリュームブロックを求めるには、  $IV_D(\alpha, d_{\max})$  を抽出すればよいことになる。同様に、case 4 の場合は  $IV_D(\alpha, d_{\max})$  と一致するので、  $IV_D(d_{\min}, \beta)$  だけを求めればよい。case 5 の場合は、サブボリューム内に  $IV_D(\alpha, d_{\max}), IV_D(d_{\min}, \beta)$  の両方の領域が含まれる。

以上の関係を用いて、求めた BONO をルートから深さ優先探索する。探索しているノードが case 1, 2 であるならば、それ以上の子ノードの探索は不要であり、次のノードを探索する。最終的に BONO のリーフに到達したとき、表 1 に従って、そのノードに含まれるすべてのキューブに対して区間型ボリュームブロックの抽出を行う。たとえば、case 3 のノードであれば、すべてのキューブに対して  $\alpha$  キューブだけを抽出し、case 5 のノードであれば、すべてのキューブに対して  $\alpha, \beta$  キューブの両方を抽出し、それらの積集合をとる。

以上のことから、図 4 のアルゴリズムが得られる。これを用いて BONO を走査すれば、高速な区間型ボリュームの抽出が可能となる。

##### 4.2 具体的な例 (二次元版)

図 4 に示したアルゴリズムを用いて、二次元の例から、  $\alpha = 25, \beta = 28$  とし、区間型ボリューム  $IV_D(25, 28)$  を抽出してみる (図 5)。

初めに、与えられたボリュームに対して、BONO を作成する (図 5 (b))。この BONO のルートに対して、BONO\_traverse を実行する。ルートに与えられたサブボリュームのフィールド値の最小値・最大値は (7, 35) であるので、case 5 の場合にあてはまり、さらに子ノードを調べる必要がある。よって、ルートの枝 0 の子を調べるとフィールド値の最小値・最大値は (7, 35) であるから、これも case 5 となり、さらに子ノードを調べる。

枝 0, 1 の子は case 1 となり、そのノードに含まれるサブボリュームは区間型ボリューム外となるので、ここで走査を終了する。枝 2, 3 の子は case 5 となるので、そのノードに含まれるサブボリューム内の各

```

procedure BONO_traverse(node)
  begin
     $I_{\max} \leftarrow \text{node.max}; I_{\min} \leftarrow \text{node.min}; \text{cube\_case} \leftarrow \text{check\_cube}(I_{\min}, I_{\max});$ 
    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  $\alpha$  cube from subvolume(node);
          break;
        case 4 : if not leaf(node) then for each child c of node do BONO_traverse(c);
          else extract  $\beta$  cube 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
  end;

```

図4 BONOを用いた区間型ボリューム高速抽出アルゴリズム

Fig. 4 A BONO-based accelerated version of interval volume extraction algorithm.

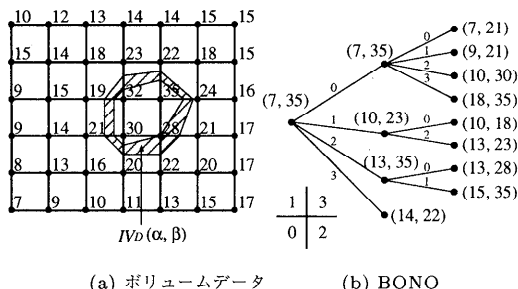


図5 二次元ボリュームデータと対応するBONOの定義

Fig. 5 A volume data (2D) and its corresponding BONO definition.

キューブに対して、区間型ボリュームを抽出する。

同様に、他のノードに対しても繰り返し、最終的に各キューブから抽出した区間型ボリュームの和集合をとれば、求める区間型ボリュームが抽出される。この例における区間型ボリュームは、図5の斜線の領域である。

従来の方法では、30個のすべてのキューブにアクセスしていたのに対し、区間型ボリュームを求める際にアクセスしたキューブは16個であった。このことから、区間型ボリュームの抽出に対してBONOを用いた場合も、等値面化にBONOを用いた場合と同等の効果が得られると期待される。

#### 4.3 BONOを用いた計測の高速化

BONOを用いることで計測も高速に行うことが可能となる。これは、区間型ボリュームが存在しないキューブへのアクセスを減少させることが可能である

ことから、無駄なキューブへのアクセスを減少させ、計算に要する時間が短縮されるためである。さらに、BONOを走査していくことにより、サブボリューム全体が区間型ボリュームに属する領域を知ることができるため、各キューブごとでなくサブボリューム自身の体積や積分値を求めることで、データへのアクセスを減少させることが可能となる。

## 5. 実行結果とその評価

前章で提案したアルゴリズムをAVS<sup>17)</sup>Ver.5上に実装した。これを用いて、さまざまなボリュームデータにおける本手法の有効性を検証する。5.1, 5.2節では、水素分子の三次元電荷密度分布を表すボリュームデータを用いて、従来の区間型ボリュームの抽出方法と、BONOを用いた抽出方法の効率の比較の他に、その他の等値面を高速に抽出する手法であるソートリストを使用する方法<sup>12)</sup>の拡張との比較を行う。続いて5.3節では、陽子・水素原子衝突における三次元電荷密度分布の時系列ボリュームデータを用いて、さらに本手法の有効性を詳細に検討する。

なお、実装・実験に使用したハードウェアはSGI社製のOnyx (CPU: R4400 × 2, Clock: 100 MHz, RAM: 128 Mbytes, 2-way interleaved)である。

#### 5.1 従来の区間型ボリューム抽出アルゴリズムとの比較

2つの水素原子は、それぞれが持つ1個の電子が互いに相手の原子の周りの軌道にもある確率で存在することで、水素分子として共有結合する。このときの

表2 区間型ボリューム生成のための所要時間  
Table 2 Timings for interval volume construction.

処理フェーズ	Regular	BONO		SORT1		SORT2	
	A	B	B/A	C	C/A	D	D/A
(1) 補助データの作成	—	0.58	—	2.78	—	88.24	—
(2) ボリュームブロックの抽出	4.14	0.58	14.01 %	1.26	30.43 %	0.45	10.87 %
(3) オブジェクトの生成	0.24	0.24	100.00 %	0.24	100.00 %	0.24	100.00 %
(4) 計測	10.48	9.20	87.79 %	9.98	95.23 %	9.64	91.98 %
(1)+(2)+(3)	4.38	1.40	31.96 %	4.28	97.72 %	88.93	2030.37 %
(2)+(3)	4.38	0.82	18.72 %	1.50	34.25 %	0.69	15.75 %

[A, B, C, Dの単位:秒]

電荷密度は二電子系の波動関数から解析的に求められることが知られている<sup>18)</sup>。この波動関数から三次元電荷密度分布を表す、解像度が  $60 \times 60 \times 60$  のボリュームを求め、それをを用いて従来の区間型ボリューム抽出アルゴリズムと BONO を用いた抽出アルゴリズムの実行時間等の比較を行った<sup>11)</sup>。

本手法の実現のために用いた AVS のボリュームデータ型では、フィールド値は1バイト符号なし型数 (0~255) に正規化される。この三次元電荷密度分布を表す区間型ボリュームの表示例を図1(b)に示した。表2に、区間 [100.00, 150.00] の区間型ボリュームの抽出に要した時間を掲げた。表における「補助データの作成」とは、区間型ボリュームを抽出するときに使用する索引データの生成に要する時間である。従来の方法では補助データを使用しないため、これに要する時間はなく、本手法の場合は BONO (動的なリスト構造) を生成する時間を指している。「ボリュームブロックの抽出」とは、各々の方法でキューブを走査し、区間型ボリュームブロックを抽出する際に要する時間である。BONO を用いた方法では、BONO を走査する時間もこれに含まれる。また、「オブジェクトの生成」とは、抽出した区間型ボリュームブロックを AVS で表示するためのジオメトリデータ型に変換するのに要する時間である。この「オブジェクトの生成」時間は、いずれの方法でも同一の区間型ボリュームブロックが抽出されるため、等しい時間を必要とする。

表2から、補助データ作成からオブジェクトの生成までの総時間で比較すると、BONO を用いた方が、より高速に区間型ボリュームを抽出できたことが分かる。さらに前述のように、一度生成した BONO は任意の区間値において再利用可能である。したがって、通常対象ボリュームの解析は指定フィールド値の変更を繰り返しながら進められることから、1つのボリュームデータから複数の区間型ボリュームを抽出する場合、BONO 生成を前処理に独立させることができ、しかもそれに要する時間はほとんど問題とならないレベル

に抑えられるといえる。

表2の例では、描画に必要な区間型ボリュームが存在するキューブは、ボリューム全体の1.76%であった。このことから、区間型ボリュームが存在するキューブはボリューム全体の一部にすぎず、すべてのキューブへのアクセスは無駄であることが分かる。この例では、アクセスしたキューブにおける無駄なキューブの割合が BONO を用いることで、98.24 (100-1.76)% から 2.56% まで減少した。この事実は、BONO が ROI の近似形状の優れた階層表現であることも示唆している。

## 5.2 他手法との比較

次に、BONO と同様に等値面を高速に抽出するアルゴリズムの1つである、ソートした索引リストを参照して等値面を抽出する方法<sup>12)</sup>を拡張して区間型ボリュームを抽出するソートリスト法と、本法との効率の比較を行った。ソートリスト法は2種類を試みた。第一のソートリスト法では、初めに各キューブにアクセスし、そのキューブの頂点ボクセルのフィールドの最小値・最大値を求める。次に、それらの最小値によって索引をソートし、索引リストを作成する。このとき、各キューブ内の最小フィールド値と最大フィールド値の差で最も大きい値  $Z$  を求めておく。区間型ボリューム  $IV_D(\alpha, \beta)$  が存在するキューブは、最小値が  $[\alpha - Z, \beta]$  に属するものである。よって、この区間に含まれる最小値を持つキューブを、索引リストから線形探索によって求めることで、区間型ボリュームが存在しない無駄なキューブへのアクセスを省くことが可能となる。

また、ソートリストを用いる第二の方法として、最小値でソートした索引リストを、同一の最小値を持つデータについて、最大値でさらにソートしたリストを作成し、それを線形探索する方法も実装した。この方法では、最小値だけでなく最大値についてもソートが行われていることから、区間型ボリュームが存在するキューブにだけアクセスすることが可能である。

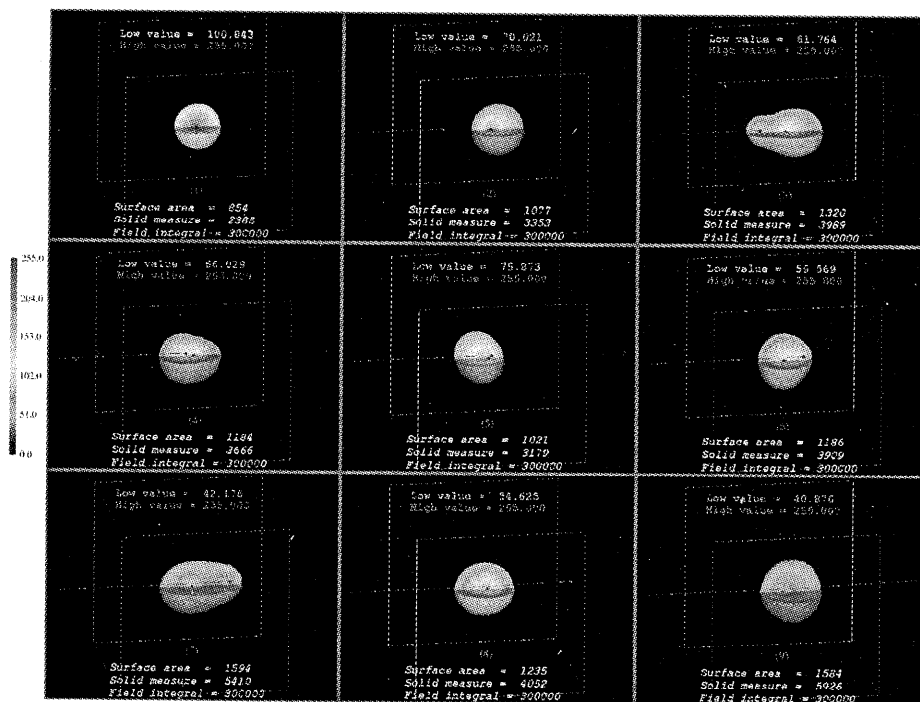


図6 陽子・水素原子衝突シミュレーションデータにおいて電子の等存在確率を示す区間型ボリューム  
 Fig. 6 Interval volumes for a single iso-electron existential probability excerpted from a proton-hydrogen atom collision animation.

表2では、前者をSORT1、後者をSORT2として、結果を示した。なお、両者ともクイックソートを実装した。SORT1では、最小値についてだけソートを行うため、SORT2と比較して索引リストの作成に要する時間が少なくすむが、その分、無駄なキューブへのアクセスも増加してしまう。SORT2では、すべてのデータを完全にソートするため、かなりの時間を必要とするが、一度索引リストを作成してしまえば、そのリストを用いることでかなり高速に区間型ボリュームを抽出することが可能である。

実験の結果、いずれの場合もソートリストの作成にかなり時間を要することが分かった。これらの方法では、解像度がある程度大きくなるとソートリストの作成に必要な膨大な時間と記憶領域が計算資源に与える影響は無視できなくなる。そこで、本手法の方がさまざまな計算環境における実装可能性の点で有効であると考えられる。

### 5.3 計測を用いた区間型ボリュームの抽出

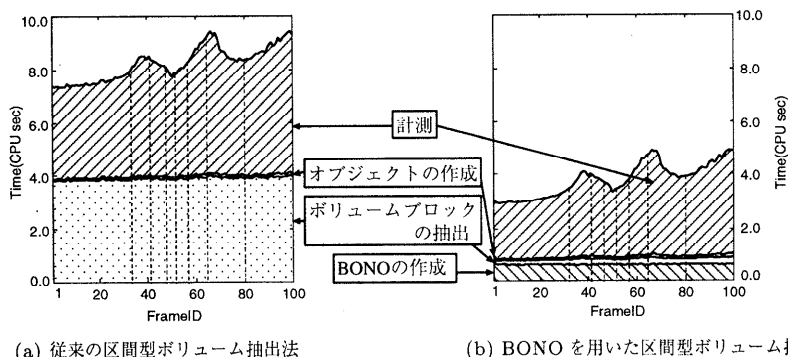
次にボリュームデータの空間コヒーレンスに対する本法の感度を調べるために、最も簡単なイオン・原子衝突系である陽子・水素原子衝突の三次元数値シミュレーションから得られた、衝突過程における電荷密度分布を表す、解像度  $60 \times 60 \times 60$  の100フレームの

時系列ボリュームデータの可視化を行った<sup>5),6)</sup>。ここでは、区間型ボリュームの計測機能を効果的に利用し、電荷密度を積分し、電子の存在確率の等しい領域、すなわち等存在確率を持つ電子雲の外向き境界を各時刻ごとに区間型ボリュームを用いて探索し、その時間変化を可視化した。これにより、入射イオンが標的原子に接近したときの電子の振舞いを直観的に知ることができる(図6)。

図7は、各時刻ごとの区間型ボリューム抽出に要した時間を、各処理フェーズごとの累積グラフで表したものである。(a)は、従来の区間型ボリューム抽出方法で要した時間を示したものであり、(b)は、本法で区間型ボリュームを抽出するのに要した時間をグラフで示したものである。図6で示した画像と対応するフレームを、グラフ中に縦の破線で示した。

このグラフから、補助データの作成からオブジェクトの生成までのトータルの時間が、BONOを用いることで約1/4に減少できたことが分かる。実際に、本手法がアクセスしたキューブに対する描画に必要な区間型ボリュームの割合は、平均0.310、標準偏差0.156であり、ボリューム全体に対する描画に必要な区間型ボリュームの割合が平均0.008、標準偏差0.016であることを考えると、無駄なキューブへのアクセス





(a) 従来の区間型ボリューム抽出法

(b) BONOを用いた区間型ボリューム抽出法

図7 ポリウムデータの空間コヒーレンスに対するBONOによる区間型ボリューム抽出アルゴリズムの図6の感度

Fig. 7 Sensitivity of interval volume extraction algorithm with/without BONO to the spatial coherence of the volume data in Fig. 6.

がかなり減っていることが分かる。

図7の抽出合計時間は、主として計測に依存しており、計測時間だけを比較すると、従来の方法では平均4.26秒、標準偏差5.53であるのに対し、本方法では平均2.90秒、標準偏差5.57であった。このことから、計測についても、従来の方法よりBONOを用いた方が高速に行えることが分かる。特に広い領域が区間型ボリュームに含まれるような場合は、BONOを用いてサブボリューム単位で計測を行うことで、かなり効率的に計測を行うことが可能となる。たとえば、 $IV_D(0, 255)$ の場合、従来の方法では計測時間が2.53秒かかっていたものが、0.4秒で行えるようになった。しかし、表からも分かるように計測に要する時間が、区間型ボリュームを抽出する時間より長くなってしまっている。これは、体積や積分値の計算がキューブを細分することで行われているため、無駄なキューブへのアクセスを減少させるだけでは十分な高速化が図れないためである。つまり、区間型ボリュームの表面積が大きくなるほど細分するキューブ数が多くなるため、空間相関性の低いデータでは計測時間が長くなる傾向があると考えられる。今後、この計測処理を改善していく必要がある。

計測を除く区間型ボリュームの抽出時間がほぼ均一であることから、区間型ボリューム抽出処理のボリュームデータコヒーレンスに対する感度は比較的低いといえる。このことから、区間型ボリュームはリアルタイムの可視化技法としての基質を有すると見なすことができる。現状では、同一の積分値を持つ区間をあらかじめ計測し、上限値と下限値を求めておき、それを参照して描画だけを行うのであれば、その処理時間に問題は無いと考えられる。

## 6. まとめと今後の課題

本論文では、与えられたボリュームデータから、フィールド値がある有限閉区間に属する三次元サブボリュームをソリッドとして表現する区間型ボリュームを、階層的なボリューム空間索引構造であるBranch-on-Need Octreeを用いて、高速に抽出する方法を提案した。また、これをAVS Ver.5上へ実装し、従来の方法やソート索引リストを用いる方法との効率を比較し、本法の効率の高さが実証された。本アルゴリズムにより区間型ボリュームはボリューム探究の基礎モデルとして有効であると結論することができる。

今回、実験に用いたデータは空間コヒーレンスが高いものであるが、一般に、ボリュームデータにおいて等値面が存在するキューブの割合は、ボリューム全体の10%前後であることが報告されている<sup>9)</sup>。このことから推測すれば、区間型ボリュームの場合も無駄なキューブへのアクセスを削減することは一般的にも有効であり、BONOを用いることでより高速な抽出が実現できると結論づけられる。

今後の課題として、以下の4点があげられる。

- 区間型ボリュームの形態的計測処理の高速化
- 細線化による等値面の高速抽出法<sup>19)</sup>などの他手法による区間型ボリュームの高速抽出の実現
- 正六面体格子構造以外のボリュームデータ構造への区間型ボリューム抽出アルゴリズムの拡張
- ボリュームデータの空間コヒーレンスの定量化とそれに応じて曖昧さの除去手法を適応的に選択することによる区間型ボリューム抽出処理の適応化

謝辞 5.3節で用いた、水素原子衝突シミュレーションの可視化という共同応用研究を通じて、数々の有益

なコメントをいただいたお茶の水女子大学理学部情報科学科の佐藤浩史教授に感謝の意を表する。

### 参考文献

- 1) Kaufman, A.E.: *Volume Visualization*, IEEE Computer Society Press (1991).
- 2) 藤代一成, 茅 暁陽, 國井利泰: ボクセル指向 3次元データ表現とその表示技術, 情報処理, Vol.34, No.3, pp.285-298 (1993).
- 3) 藤代一成, 前田裕治: Marching Cubes の一般化: 高解像度区間型ボリューム抽出アルゴリズムの提案, 情報処理学会研究会報告, 94-CG-71-5, pp.25-31 (1994).
- 4) 前田裕治, 藤代一成, 竹島由里子: Solid Fitting を用いたボリュームデータの可視化, グラフィクスとCADシンポジウム論文集, pp.33-40 (1995).
- 5) Fujishiro, I., Maeda, Y. and Sato, H.: Interval Volume: A Solid Fitting Technique for Volumetric Data Display and Analysis, *Proc. IEEE Visualization '95*, pp.151-158 (1995).
- 6) Fujishiro, I., Maeda, Y., Sato, H. and Takeshima, Y.: Volumetric Data Exploration Using Interval Volume, *IEEE Trans. Visualization and Computer Graphics*, Vol.2, No.2, pp.144-155 (1996).
- 7) Guo, B.: Interval Set: A Volume Rendering Technique Generalizing Isosurface Extraction, *Proc. IEEE Visualization '95*, pp.3-10 (1995).
- 8) Crawfis, R.A.: Real-time Slicing of Data Space, *Proc. IEEE Visualization '96*, pp.271-277 (1996).
- 9) Wilhelms, J. and Van Gelder, A.: Octrees for Faster Isosurface Generation, *ACM Trans. Graphics*, Vol.11, No.3, pp.201-227 (1992).
- 10) 竹島由里子, 前田裕治, 藤代一成: 区間型ボリューム抽出アルゴリズムの高速化, 第50回情報処理学会全国大会論文集, 1C-10 (1995).
- 11) 竹島由里子: 対話的ソリッドフィッティングのための区間型ボリューム抽出アルゴリズムの高速化, 修士論文, お茶の水女子大学大学院理学研究科 (1996).
- 12) Giles, M. and Haines, R.: Advanced Interactive Visualization for CFD, *Computing System in Engineering*, Vol.1, No.1, pp.51-62 (1990).
- 13) Mäntylä, M.: *An Introduction to Solid Modeling*, Computer Science Press (1988).
- 14) Lorensen, W.E. and Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics*, Vol.21, No.4, pp.163-169 (1987).
- 15) Nielson, G.M. and Hamann, B.: The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes, *Proc. IEEE Visualization '91*,

pp.83-90 (1991).

- 16) 竹島由里子, 藤代一成, 山下優子: 区間型ボリュームにおける位相的あいまいさの緩和, 情報処理学会研究会報告, 97-CG-84-5, pp.25-32 (1997).
- 17) Advanced Visual Systems Inc.: *AVS Developer's Guide* (1992).
- 18) Wang, S.: The Problem of the Normal Hydrogen Molecule in the New Quantum Mechanics, *Physical Review*, Vol.31, pp.579-586 (1928).
- 19) Itoh, T., Yamaguchi, Y. and Koyamada, K.: Volume Thinning for Automatic Isosurface Propagation, *Proc. IEEE Visualization '96*, pp.303-310 (1996).

(平成8年11月28日受付)

(平成9年6月3日採録)



藤代 一成 (正会員)

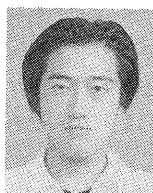
昭和35年生。昭和60年筑波大学大学院博士課程工学研究科電子・情報工学修士号取得退学。同年東京大学理学部情報科学科助手。その後、筑波大学電子・情報工学系を経て、

平成3年よりお茶の水女子大学理学部情報科学科助教、現在に至る。理学博士(東京大学)。ビジュアルコンピューティング、特に、ボリュームビジュアルゼーション/グラフィクスとその応用に興味を持つ。著書に「CAD/CAM」(丸善)他。IEEE Computer Society, ACM等の会員。情報知識学会理事。



竹島由里子 (学生会員)

昭和46年生。平成8年お茶の水女子大学大学院理学研究科情報科学専攻修了。同年同大学院人間文化研究科人間環境学専攻博士課程入学。現在、在学中。ボリュームビジュアルゼーションやインフォメーションビジュアルゼーションに興味を持つ。



前田 裕治 (正会員)

昭和46年生。平成5年筑波大学第三学群情報学類卒業。平成7年同大学院工学研究科博士前期課程修了。平成8年同博士後期課程中退。同年福井県庁入庁。Volume visualization, Biomedical imaging と Information visualization に興味を持つ。