Vol.2012-CG-149 No.20
Vol.2012-CVIM-184 No.20
2012/12/4

非一様弾性体の対話的変形シミュレーション

高岡 和史[1,a)]　藤代 一成[2]

概要：コンピュータグラフィクスの発達により，大変形を伴う変形シミュレーションをリアルタイムに表現できるようになった．しかしその大半が均質体の変形や均質体同士の衝突計算であり，非均質な塑性をもつ物体の変形を扱った研究はほとんど報告されていない．そこで本稿では，ボリュームモデリングと弾性体変形手法を組み合わせ，非一様な塑性分布を考慮した新たな物体変形シミュレーション手法として，FastLSM+を提案する．Constructive Volume Geometry によって塑性分布を作成し，高速な弾性体変形手法である Fast Lattice Shape Matching に拡張した塑性分布を反映させることで，これまで困難であった任意の形状と塑性分布をもつ物体の変形シミュレーションを実現する．

キーワード：弾性体，変形，塑性分布，ボリュームオブジェクト，Constructive Volume Geometry，形状マッチング

Interactive Deformation of Non-Uniform Elastic Bodies

Kazufumi Takaoka[1,a)]　Issei Fujishiro[2]

Abstract: Although real-time large deformation simulation has recently been reported in the computer graphics literature, most of the existing approaches handle only the deformation of homogeneous bodies and the collision among homogeneous bodies. In this article, therefore, we present a new simulation framework for non-uniform elastic object deformation using volume modeling and fast deformation methods. First, we design a stiffness map by utilizing constructive volume geometry, which is known as an algebraic scheme for modeling complex volumetric objects using combinational operations. Then, a new deformation method called FastLSM+, which is a modified Fast Lattice Shape Matching so as to reflect the effects from the stiffness map, computes how the object deforms with the designed stiffness map. This framework enables ones to simulate the deformation of objects with arbitrary shape and stiffness distribution.

Keywords: Elastic body, deformation, stiffness map, constructive volume geometry, shape matching.

1. 序論

近年コンピュータグラフィックス（CG）において形状変化に関する研究が進み，リアルタイムレンダリングを必要とする作品においても，本来計算負荷の高いやダイナミックな形状変化を伴う物体を表現できるようになった．しかし，その大半が均質体の形状変化や複数の独立した物体間の衝突を扱ったものばかりであり，人体や食物のように複雑な構造や非均質な塑性分布をもつ物体の変形や，それに伴う接触を扱った研究はほとんど報告されていない．

これは，複雑な内部構造を表現するためにはボリュームデータが必要であり，ボリュームデータが次の二つの問題を抱えていることが原因であると考えられる．一つ目は計算コストの高さである．ボリュームデータはデータ量がソリッドモデルに比べて大きく，それに伴って計算コストも高くなり，従来のハードウェアでは実時間性を確保できなかった．そして二つ目はボリュームデータの入手難易度の高さである．以前は，CT や MRI といった高額な機材を用いなければ任意のボリュームデータを取得できなかった．

[1] 慶應義塾大学　大学院理工学研究科
Graduate School of Science and Technology, Keio University
[2] 慶應義塾大学　理工学部情報工学科
Department of Information and Computer Science, Faculty of Science and Technology, Keio University
[a)] takaoka@fj.ics.keio.ac.jp

このように，ボリュームデータの利用機会が増えてきているが，依然として解決すべき問題も多く，研究対象として注目を集め続けている．

ボリュームデータは現在主流となっているソリッドモデルよりも多くの情報をもち，形状操作やデータ操作に関しても直感的でわかりやすく，計算コストを考えなければソリッドモデルよりも多くの点で優れているデータ形式である．将来的には，ボリュームデータによるモデリングやシミュレーションが主流になると予想されるため，ボリュームデータを用いた高速なシミュレーション手法が必要となる．そのため，ボリュームデータを用いた対話的なシミュレーションの実現は，CG 技術が発達した現在においても重要な研究分野の一つである．

そこで本研究では，塑性分布を与えるボリュームデータの作成から変形までを一貫して行う，非均質弾性体の変形シミュレーションシステムを提案する．

ボリュームデータの作成には，物体間の関係を代数演算で定義することで任意形状を定義できる Constructive Volume Geometry（CVG）[1] を採用し，簡単な操作によるボリュームデータ作成を実現する．CG シミュレーションでは物理的正確さの向上が求められる一方で，コンピュータゲーム等の世界では頑健性と速度の方がしばしば優先される．そこで，物理的正確さを部分的に犠牲にしつつも，頑健で高速な弾性体変形手法の Fast Lattice Shape Matching（FastLSM）[2] が開発された．本研究では FastLSM を塑性分布の影響を反映するように拡張し，新たな非均質物体の変形手法 FastLSM+として，複雑な物体の変形計算を高速化する基盤を構築する．

本稿は次節以降，以下のように構成されている．まず次節で弾性体変形とボリュームモデリングに関する関連手法を挙げ，3 節でボクセル化処理，CVG によるボリュームモデリング，FastLSM による変形計算といった本研究の特徴を述べる．4 節で本研究の評価を行い，5 節で本稿をまとめ，今後の課題に言及する．

2. 関連研究

関連研究として，2.1 項で高速な弾性体変形手法であるシェイプマッチングについて述べ，2.2 項で塑性分布作成に必要なボリュームモデリングについて述べる．

2.1 弾性体変形シミュレーション

大変形を伴う対話的シミュレーションは以前から CG における重要な研究対象である．しかし，未だに大変形シミュレーションをリアルタイムに行えるアプリケーションは無く，高速な手法があったとしても何らかの制限下でしか動作しない．加えて，そういった手法の多くはとても複雑で実装が難しいため，より高速かつ簡単な手法が求められている．

Müller らが提案したシェイプマッチング [3] は，形状マッチングによる弾性体変形手法である．シェイプマッチングは，対応する頂点間の変形距離の総和を最小にするように働く力で弾性体を表現した．物理的正確さは無いが，基準座標と現在の座標の差分を計算するだけという単純かつ高速な手法なため，現在は様々な対象に応用されている汎用的な弾性体変形手法である．本研究で用いる FastLSM[2] は，Rivers らが Müller らの手法を高速化したものである．FastLSM は物体を粗い格子の中に埋め込み，その粗い格子に対してシェイプマッチングを実行することで，計算コストを抑えながらも高解像度なメッシュの大変形をリアルタイムに実現している．

さらに，FastLSM には別の高速化の工夫も行われている．シェイプマッチングは元々軟らかい物体表現のための手法であるが，形状マッチングを行う領域を広くとることで剛体を表現できる．しかしながらマッチング領域の重複が計算効率を下げてしまっていた．Rivers らは，総和計算が領域依存ではなく格子点にのみ依存することを発見し，独自の総和アルゴリズムを考案した．この総和計算は計算済みの値を最大限に再利用することで，マッチング領域の一辺の粒子数を n としたとき，計算量を $O\left(n^3\right)$ から $O\left(1\right)$ へと下げることに成功し，計算速度を飛躍的に向上させている．

本研究では，作成した塑性分布と粗い格子を対応付けることにより，物体の局所的な変形を実現する．詳しくは 3 節で述べる．

2.2 ボリュームモデリング

3 次元形状モデリングは，常に CG 研究において主要な研究分野である．従来のモデリング手法は，ポリゴンに代表される表面幾何形状を基に作られるソリッドモデルが一般的であった．この表現手法は人工物などを表現するのに優れており，現在の CG では主流となっている．しかしながら，ソリッドモデルでは物体の内外判定は表現できるが，物体内部が均一であると仮定してモデリングしている．このため，複雑な内部構造を考慮した変形計算は困難であった．一方，ボリュームデータは物体の内部の値まで定義できるため，人体や地層といった複雑な情報をもつことができる．しかし MRI や CT，3D スキャナといった専用の入力デバイスや高価で専門的なソフトウェアを利用しなければ作成することができず，一般の人が任意の形状をもつボリュームデータを容易に取得することはできなかった．この問題を解決するために，Min Chen らはソリッドモデリング手法である Constructive Solid Geometry（CSG）を，ボリュームモデリングに適応させた Constructive Volume Geometry（CVG）[1] を開発した．本研究では，塑性分布の作成に CVG を用いることで，複雑な内部構造の作成を自由に行えるようにする．

Vol.2012-CG-149 No.20
Vol.2012-CVIM-184 No.20
2012/12/4

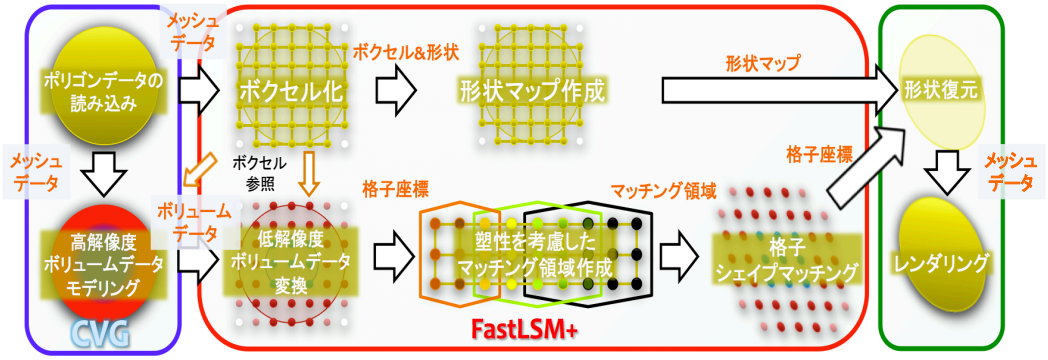

図 1 提案システムにおける処理フロー

3. アプローチ

本研究のアプローチについて述べる．FastLSM[2] をもとに，CVG[1] によって作成した塑性分布による変形制御を追加し，任意の内部構造を考慮した高速な変形シミュレーションを実現する．本研究のフレームワークは，前処理の塑性分布作成と，主処理の変形シミュレーション，後処理のレンダリングという 3 つの処理から構成される（**図 1**）．

3.1 塑性分布の作成

3.1.1 基準形状の読み込みとボクセル化

本システムでは解像度の異なる 2 種類のボリュームデータが必要となる．

一つ目は，オリジナルの塑性分布となる高解像度ボリュームデータである．まず，外部ソフトウェアでポリゴンデータを作成したものをベース形状とし，そこに CVG を用いて詳細な塑性を付加して作成する．ボリュームデータへ変換するボクセル化手法には，フレームバッファのブレンディング関数を利用して 3D オブジェクトのスライス画像を生成し，それらから 3D テクスチャマッピングの機能を用いてボクセルモデルを作成する方法 [4] を用いる．

二つ目は，高解像度ボリュームデータを基に低解像変換したボリュームデータである．低解像変換は単純に周囲のデータの平均をとるが，このとき硬さに重み付けをして硬い部分が際立つよう調整する．また，同時に変形後の格子データからポリゴン形状を復元をするための形状マップも作成する．格子点とポリゴンデータの各頂点座標を 3 重線形補間によって対応付け，頂点リストとして保持する．

3.1.2 Constructive Volume Geometry

CVG は，物体同士の関係を CVG 代数とよばれる簡単な代数演算により指定することで，複雑なボリュームデータを構築する手法である（**図 2**）．CVG 代数は空間オブジェクトといくつかの演算子により表される，以降，実数全体の集合を $\mathbb{R}$，三次元ユークリッド空間を $\mathbb{E}^3$，スカラ場は関数 F を用いて $F : \mathbb{E}^3 \to \mathbb{R}$ で与えられる．

まず，空間オブジェクト $\mathbf{o}$ は次のように定義される．

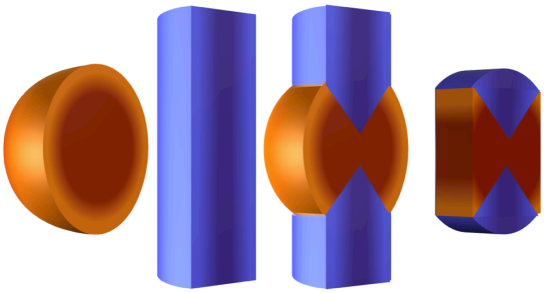

図 2 CVG によるオブジェクト同士の合成結果の断面図．左から順番に「球」「円柱」「球と円柱の和」「球と円柱の積」を表す

$$\mathbf{o} = (O, A_1, \ldots, A_k)$$

ここで $\mathbf{o}$ は該当するオブジェクトを表し，$O : \mathbb{E}^3 \to [0, 1]$ はオブジェクトの可視性を表す．$A_1, \ldots, A_k$ はオブジェクトの特性値を意味しており，必要な個数だけ定義できる．本研究では，特性値として色 R, G, B と塑性 S を与え，空間オブジェクト $\mathbf{o}$ を次のように定義する．

$$\mathbf{o} = (O, R, G, B, S)$$

本システムでは基本となる空間オブジェクトとして球，立方体，円柱，円錐を提供する．

続いて，本研究では複雑な形状を生成するために，以下の 6 つの演算子を用いる．まずは和（union），積（intersection），差（difference）である．この 3 つは CSG でも用いられており，集合演算による表面形状表現には必要不可欠な演算子である．次に，物体内部を編集するための演算子として混合（blend）を用い，特定の値を取得するためにはキャプ（cap）とトリム（trim）を用いる．

- *union* : $\boxed{\cup}(\mathbf{o}_1, \mathbf{o}_2) = (\mathbf{MAX}(O_1, O_2),$
  $\mathbf{SELECT}(O_1, R_1, O_2, R_2), \mathbf{SELECT}(O_1, G_1, O_2, G_2),$
  $\mathbf{SELECT}(O_1, B_1, O_2, B_2), \mathbf{MAX}(S_1, S_2))$
- *intersection* : $\boxed{\cap}(\mathbf{o}_1, \mathbf{o}_2) = (\mathbf{MIN}(O_1, O_2),$
  $\mathbf{SELECT}(O_1, R_1, O_2, R_2), \mathbf{SELECT}(O_1, G_1, O_2, G_2),$
  $\mathbf{SELECT}(O_1, B_1, O_2, B_2), \mathbf{MIN}(S_1, S_2))$
- *difference* : $\boxed{-}(\mathbf{o}_1, \mathbf{o}_2) =$
  $(\mathbf{SUB}_{01}(O_1, O_2), R_1, G_1, B_1, S_1)$
- *blend* : $\boxed{\oplus}(\mathbf{o}_1, \mathbf{o}_2) = (\mathbf{ADD}_{01}(O_1, O_2),$

Vol.2012-CG-149 No.20
Vol.2012-CVIM-184 No.20
2012/12/4

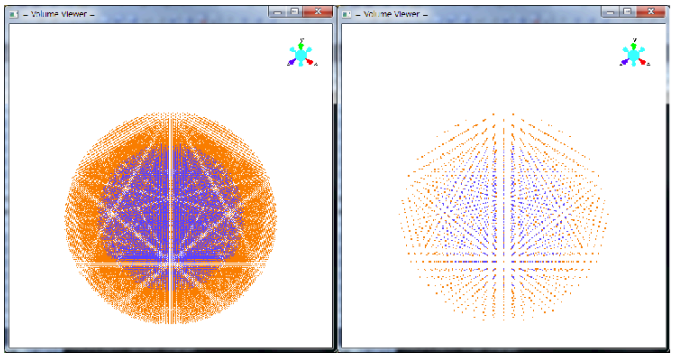

図 3 作成したボリュームマップ (左) と変形計算に用いる格子データ (右) の比較．硬さによって色分けし，3 次元空間上の格子点を表示している

$\mathbf{MIX}_{01}(O_1, R_1, O_2, R_2), \mathbf{MIX}_{01}(O_1, G_1, O_2, G_2),$
$\mathbf{MIX}_{01}(O_1, B_1, O_2, B_2), \mathbf{ADD}_{01}(S_1, S_2))$

- $cap : \boxed{\triangle}(\mathbf{o}_1, \mathbf{o}_2) = (\mathbf{CAP}_{01}(O_1, O_2),$
  $\mathbf{CAP}_{01}(R_1, R_2), \mathbf{CAP}_{01}(G_1, G_2),$
  $\mathbf{CAP}_{01}(B_1, B_2), \mathbf{CAP}_{01}(S_1, S_2))$
- $trim : \boxed{\bigtriangledown}(\mathbf{o}_1, \mathbf{o}_2) = (\mathbf{TRIM}_{01}(O_1, O_2),$
  $\mathbf{TRIM}_{01}(R_1, R_2), \mathbf{TRIM}_{01}(G_1, G_2),$
  $\mathbf{TRIM}_{01}(B_1, B_2), \mathbf{TRIM}_{01}(S_1, S_2))$

各種演算の詳細は付録 A を参照されたい．

一般に，高解像度なボリュームデータは，データ量が大き過ぎるためにタイムステップごとに交差判定やデータの更新処理を行うことが難しい．しかし CVG が必要とするのはオブジェクトがもつボリュームデータではなく，ワールド座標系における現在の座標 $\mathbf{x}$ と回転行列 $\mathbf{R}$，合成相手のオブジェクト番号 i_{obj} とその演算関係を示すパラメータ $\mathbf{TYPE}_{mix}$ のみでよく，ボリュームデータの更新処理を常に行う必要がない．また，更新の際もオブジェクト $\mathbf{o}_1$ と $\mathbf{o}_2$ の座標からその重複部分をバウンディングボックスを用いて判定し，重複部分のデータについてのみ与えられた演算を行うだけでよく，すべてのデータを更新する必要はない．したがって複雑なボリュームオブジェクトであっても高速に作成することができる．

3.2 変形シミュレーション

本節では，本稿の主要部分である塑性を考慮した高速な変形計算手法 FastLSM+について述べる．

3.2.1 粗い格子生成と形状マップ作成

まず，高解像度ボリュームデータを変換して，FastLSM による変形計算用の低解像度ボリュームデータを生成する（**図 3**）．この変換には 3 重線形補間を用いて格子点付近の塑性を決定する．また解像度の低下具合は必要に応じて任意に決められるものとする．同時に，元形状がポリゴンデータの場合，各ボリュームデータとの対応マップを作成しておき，変形結果からの復元に用いる．

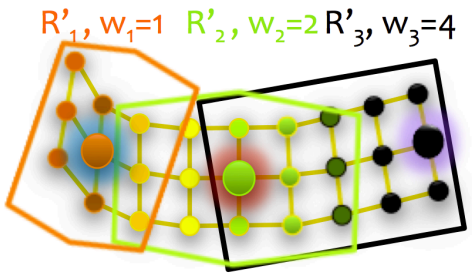

図 4 伝達距離 w_1, w_2, w_3 とマッチング領域 $\mathcal{R}'_1, \mathcal{R}'_2, \mathcal{R}'_3$ の変形への影響

3.2.2 マッチング領域

格子シェイプマッチング（LSM）は，高解像度なメッシュの代わりに，粗いボクセルによって計算することで高速化する手法である．埋め込まれたメッシュの頂点とボクセルの中心点を 3 重線形補間で対応づけることでボクセルの変形をメッシュへ反映する．このときボクセルの変形は，ボクセルの各頂点に配置された質点粒子によって制御される．各格子の頂点と粒子は同じインデックス i を共有し，粒子 i は初期位置 $\mathbf{x}_i^0$，動的位置 $\mathbf{x}_i$，そして質量 m_i によって表現される．さらに粒子 i と隣接する格子情報が含まれる近傍リスト $\mathcal{N}_i$ を構成し，近傍探索や総和計算の高速化を行う．近傍リスト作成後，マッチング領域 $\mathcal{R}_i$ に存在する粒子に対してシェイプマッチングを行う．$\mathcal{R}_i$ は粒子 i から伝達距離 w の範囲内に存在している近傍リストである．例えば $w = 1$ ならば $\mathcal{R}_i = \mathcal{N}_i$ である．

3.3 塑性を反映したマッチング領域の作成

Rivers らの実装 [2] では，伝達距離 w は物体ごとに 1 つだけ定義される定数だったため，w を変化させても物体全体の硬さが変化するだけで局所的に違う変形は実現できていなかった．しかし LSM において，物体の硬さを決める最大の要因はこの w である．具体的には w の値が大きいほど変形の挙動が剛体に近くなり，逆に w の値が小さければより軟らかな物体を表現できる．これは w が大きいと広い範囲の近傍リストを参照することになり，そこに所属する粒子の重みの平均値をとるため，周囲の粒子に対し粒子の一つあたりの移動量が相対的に小さくなるからである．

そこで本研究では，この塑性に大きく影響を与える w に注目した．Rivers らとは異なり，物体全体として一つの値をもつのではなく，各粒子に固有の伝達距離 w_i をもたせる．その w_i の値を塑性分布から算出することで，粒子 i ごとに広さの異なるマッチング領域 $\mathcal{R}'_i$ を定義でき，局所的に硬さの違う変形を実現する（**図 4**）．

3.3.1 塑性を反映した格子シェイプマッチング

ここでは，塑性を反映した格子シェイプマッチングの具体的なアルゴリズムを示す．ここで，前項で述べた塑性を反映したマッチング領域 $\mathcal{R}'_r$ は既に計算済みであるとする．

Vol.2012-CG-149 No.20
Vol.2012-CVIM-184 No.20
2012/12/4

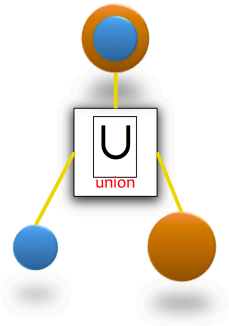

図 5 図 6 の図形の組合せを表した CVG 木の例．青色の球は半径 0.4, 硬さ 1.0．オレンジ色の球は半径 0.8, 硬さ 0.2

まず，各領域 r における粒子 i の初期座標 $(\mathbf{x}_i^0)_{i\in\mathcal{R}'_r}$ と動的座標 $(\mathbf{x}_i)_{i\in\mathcal{R}'_r}$ を一致させるような剛体変換行列 $\mathbf{T}_r$ を得る．$\mathbf{T}_r$ を得るまでの流れは，静止位置の回転行列 $\mathbf{R}_r$，領域 r の質量中心座標 $\mathbf{c}_r$ と初期質量中心座標 $\mathbf{c}_r^0$，領域 r の重み付け和 $\mathbf{M}_r$ を用いて次のよう定義される．このとき，粒子は多くの領域に属しているため，それ以外の領域からの重み付けの影響を受けないようにするために，使用する質量は $\tilde{m}_i = m_i/|\mathcal{R}_i|$ とする．$|\mathcal{R}_i|$ は領域 r に存在する粒子の個数を表す．

$$
\begin{aligned}
\mathbf{T}_r &= [\mathbf{R}_r \ \ (\mathbf{c}_r - \mathbf{R}_r\mathbf{c}_r^0)] \in \mathbb{R}^{3\times 4} \\
\mathbf{c}_r &= \frac{1}{\mathbf{M}_r}\sum_{i\in\mathcal{R}_r}\{\tilde{m}_i\mathbf{x}_i\} \\
\mathbf{c}_r^0 &= (\sum_{i\in\mathcal{R}_r}\tilde{m}_i\mathbf{x}_i^0)/\mathbf{M}_r \\
\mathbf{M}_r &= \sum_{i\in\mathcal{R}_r}\tilde{m}_i
\end{aligned}
$$

ここで使われる回転行列 $\mathbf{R}_r$ は，変形勾配テンソル $\mathbf{A}_r$ の回転要素 $\mathbf{A}_r = \mathbf{R}_r\mathbf{U}_r$ として表せる．$\mathbf{U}_r$ は 3×3 の直交行列であるため，ヤコビ法を使い回転ん行列 $\mathbf{R}_r = \mathbf{A}_r\mathbf{U}_r^{-1}$ を得る．$\mathbf{A_r}$ は次式で定義されている．

$$
\mathbf{A}_r = \sum_{i\in\mathcal{R}_r}\{\tilde{m}_i\mathbf{x}_i\mathbf{x}_i^{0\mathrm{t}}\} - \mathbf{M}_r\mathbf{c}_r\mathbf{c}_r^{0\mathrm{t}}
$$

最後に，粒子の速度 $\mathbf{v}_i(t+h)$ と最終座標 $\mathbf{x}_i(t+h)$ は，Müller ら [3] によれば目標座標の平均値 $\mathbf{g}_i = <\mathbf{T}_r\mathbf{x}_i^0>_{r\in\mathcal{R}_r}$ を用いて次のように定義される．

$$
\begin{aligned}
\mathbf{v}_i(t+h) &= \mathbf{v}_i(t) + \frac{\mathbf{g}_i(t)-\mathbf{x}_i(t)}{h} + h\frac{\mathbf{f}_{ext}(t)}{m_i} \\
\mathbf{x}_i(t+h) &= \mathbf{x}_i(t) + h\mathbf{v}_i(t+h)
\end{aligned}
$$

3.4 レンダリング

FastLSM の変形結果は低解像度ボリュームデータである．よって，そこから高解像度ボリュームデータもしくは元形状のポリゴンデータを復元してレンダリングする必要がある．これは予め作成しておいた対応マップを用いることで容易に元形状を復元することができる．復元後は通常通りメッシュを扱えばよいため，任意のレンダリング処理が行える．本稿では，物体内部の塑性をわかりやすくみせるために，格子シェイプマッチングに利用した格子点を表示した．

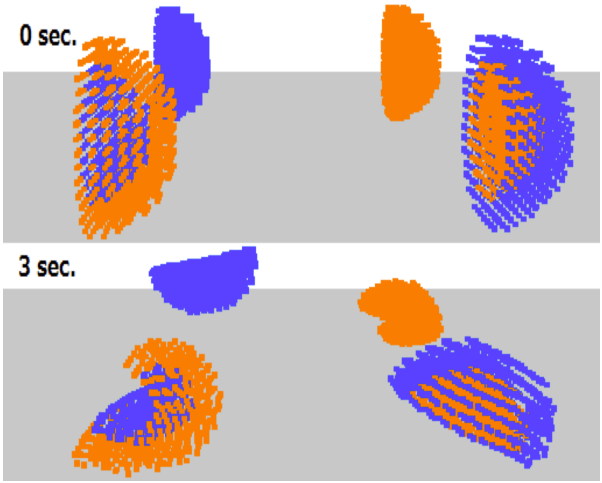

図 6 塑性を考慮した変形の例．左側手前の半球の CVG 木を**図 5** に示す．シミュレーション開始時 (上) と開始から 3 秒後の状態 (下)．硬い部位は青色，軟らかい部位はオレンジ色で描画

4. 結果と評価

半径の異なる 2 種類の半球（それぞれ半径 0.4 と半径 0.8）を用意し，硬い部分の塑性を 1.0，軟らかい部分の塑性を 0.2 としてそれぞれを異なる塑性分布になるように 4 つのオブジェクトを作成した．そのうち非均質な塑性をもつオブジェクトの CVG 木の例を**図 5** に示す．そして自由落下シミュレーションを行ったところ，それぞれ変形の様子が異なっていた．その様子を**図 6** に示す．このことから塑性分布の作成からそれを反映した変形計算が実現できていると言える．

しかしながら，どの物体も全体的に軟らかく，表現できる硬さの範囲は小さいと言える．

5. 結論と今後の課題

5.1 結論

最近注目を集め始めているボリュームデータによる変形シミュレーションに着目し，これまで殆どみられなかった任意の形状と塑性分布をもつ物体の高速な変形シミュレーションシステムの構築を行った．ユーザは外部ソフトウェアによって設計した任意形状のデータを入力し，CVG によって自由に塑性分布を作成する．そして高速な弾性体変形手法である FastLSM に塑性が反映するよう拡張することで，任意の内部構造の作成から変形シミュレーションまでを一貫して行える非一様弾性体の変形シミュレーションシステムの実行環境が提供されている．

5.2 今後の課題

現時点では，局所的な塑性は変形に反映しているが変形の様子が全体的に均質体に近く，硬い部分と軟らかい部分の差異を十分に表現しきれていない．格子の解像度を上げれば変形の正確さは増すが，一方で計算コストも増加してしまうため，別の解決策を模索する必要がある．これは，マッチング領域を伝達距離 w_i によって均一に取得していることが原因と考えられ，この点については伝達距離 w_i と伝達関数 $T_{width}(w_i)$ によって方向性をもたせたマッチング領域を生成することで回避可能と考えられる．

また，物体の変形において衝突計算を無視することはできない項目であるが，現時点では物体同士の衝突計算を考慮していない．しかし，変形計算に利用した格子点を衝突計算に使用するには，格子点の個数が多過ぎる．よって，衝突判定には Müller ら [6] を参考に，異方性粒子を利用が有効だと考えられる．物体を覆った格子に内接するように異方性粒子を適切に配置することで，計算に用いる粒子数を必要最小限に抑えることができ，衝突時に発生する力のモーメントを計算することで計算要素数が減少しても衝突計算の正確さを維持することができる．

全く塑性が変化しない物体も非現実的であるため，塑性に時間変化を与えてより複雑な変形を実現することも考えられる．本手法はパラメータの拡張は容易であるが，FastLSM はマッチング領域が一定であるため高速化を可能にしている．そこでマッチング領域の大幅な改変が頻発した場合は計算速度が遅くなる可能性がある点は示しておく必要がある．

塑性分布の作成時には，物体の内部を可視化するためにボリュームレンダリングを行う必要がある．このとき，物体の硬さを可視性の閾値に設定することで，モデリング効率の改善につながる．変形シミュレーション時には，モデリング時と同様に内部を表示することはデータ量の関係上難しいため，3 重線形補間で形状復元したサーフェスモデルにレンダリングを用いる必要がある．

最後に高速化について検討する．スライス画像は GPU のテクスチャ処理との親和性が高く，また各スライス画像は独立しているために並列処理も行える．GPU の並列演算を利用することで，さらに大規模なボリュームデータの作成がより手軽にできるようになる．

参考文献

[1] Chen, M. and Tucker, V.: “Constructive Volume Geometry,” *Computer Graphics forum*, Vol. 19, No. 4, pp. 281-293 (March 2000).
[2] Rivers, A. and James, D.: “FastLSM: Fast Lattice Shape Matching for Robust Real-Time Deformation,” *ACM Transactions on Graphics*, Vol. 26, No. 3, Article 82 (July 2007).
[3] Müller, M., Heiderberger, B., Teschner, M. and Gross, M.: “Meshless Deformations Based on Shape Matching,” *ACM Transactions on Graphics*, Vol. 24, No. 3 (July 2005).
[4] Fang, S. and Chen, H.: “Hardware Accelerated Voxelization,” *Computers and Graphics*, Vol. 24, No. 3, pp. 433-442 (June 2000).
[5] Requicha, A. A. G.: “Representations for Rigid Solids: Theory, Methods, and Systems,” *ACM Computing Surveys*, Vol. 12, No. 4, pp. 437-464 (December 1980).
[6] Müller, M. and Chentanez, N.: “Solid Simulation with Oriented Particles,” *ACM Transactions on Graphics*, Vol. 30, No. 4, Article 92 (July 2011).

付　　録

A.1 スカラ演算命令

CVG におけるスカラ演算は以下のように定義する．スカラ値 $s \in [0,1]$ については $\mathbf{ADD}_{01}$，$\mathbf{SUB}_{01}$，$\mathbf{DIV}_{01}$，$\mathbf{MIX}_{01}$，$\mathbf{CAP}_{01}$，$\mathbf{TRIM}_{01}$ を用いる．

$$\mathbf{MAX}(s_1,s_2) = \begin{cases} s_1 & s_1 \geq s_2 \\ s_2 & s_1 < s_2 \end{cases}$$

$$\mathbf{MIN}(s_1,s_2) = \begin{cases} s_1 & s_1 \leq s_2 \\ s_2 & s_1 > s_2 \end{cases}$$

$$\mathbf{ADD}(s_1,s_2) = s_1 + s_2$$

$$\mathbf{SUB}(s_1,s_2) = s_1 - s_2$$

$$\mathbf{MULT}(s_1,s_2) = s_1 \times s_2$$

$$\mathbf{DIV}(s_1,s_2) = s_1 \div s_2$$

$$\mathbf{SELECT}(s_1,t_1,s_2,t_2) = \begin{cases} t_1 & s_1 \geq s_2 \\ t_2 & s_1 < s_2 \end{cases}$$

$$\mathbf{MIX}(s_1,t_1,s_2,t_2) = \begin{cases} \frac{t_1|s_1|+t_2|s_2|}{|s_1|+|s_2|} & |s_1|+|s_2| \neq 0 \\ (t_1+t_2)/2 & |s_1|+|s_2| = 0 \end{cases}$$

$$\mathbf{CAP}(s_1,s_2) = \begin{cases} s_1 & s_1 \leq s_2 \\ -\infty & s_1 > s_2 \end{cases}$$

$$\mathbf{TRIM}(s_1,s_2) = \begin{cases} s_1 & s_1 \geq s_2 \\ -\infty & s_1 < s_2 \end{cases}$$

$$\mathbf{ADD}_{01}(s_1,s_2) = \mathbf{MIN}(1, s_1+s_2)$$

$$\mathbf{SUB}_{01}(s_1,s_2) = \mathbf{MAX}(0, s_1-s_2)$$

$$\mathbf{DIV}_{01}(s_1,s_2) = \mathbf{MAX}(0, \mathbf{MIN}(s_1-s_2))$$

$$\mathbf{MIX}_{01}(s_1,t_1,s_2,t_2) = \begin{cases} \frac{t_1s_1+t_2s_2}{s_1+s_2} & s_1+s_2 \neq 0 \\ (t_1+t_2)/2 & s_1 = s_2 = 0 \end{cases}$$

$$\mathbf{CAP}_{01}(s_1,s_2) = \begin{cases} s_1 & s_1 \leq s_2 \\ 0 & s_1 > s_2 \end{cases}$$

$$\mathbf{TRIM}_{01}(s_1,s_2) = \begin{cases} s_1 & s_1 \geq s_2 \\ 0 & s_1 < s_2 \end{cases}$$